US006125445A

United States Patent [19]

Arditti et al.

[11] Patent Number: 6,125,445

[45] Date of Patent: Sep. 26, 2000

[54] **PUBLIC KEY IDENTIFICATION PROCESS USING TWO HASH FUNCTIONS**

[75] Inventors: **David Arditti**, Clamart; **Henri Gilbert**, Bures sur Yvette; **Jacques Stern**, Paris; **David Pointcheval**, Cachan, all of France

[73] Assignee: **France Telecom**, Paris, France

[21] Appl. No.: **09/076,818**

[22] Filed: **May 13, 1998**

[30] **Foreign Application Priority Data**

May 13, 1997 [FR] France 97 05830

[51] **Int. Cl.**[7] **H04L 9/00**

[52] **U.S. Cl.** **713/169**; 713/168; 713/200

[58] **Field of Search** 380/28; 713/168, 713/169, 176, 200

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,995,082	2/1991	Schnorr	380/23
5,140,634	8/1992	Guillou et al.	380/23
5,218,637	6/1993	Angebaud et al.	380/23
5,323,146	6/1994	Glaschick	340/825.34
5,502,764	3/1996	Naccache	380/23
5,790,667	8/1998	Omori et al.	380/23

FOREIGN PATENT DOCUMENTS

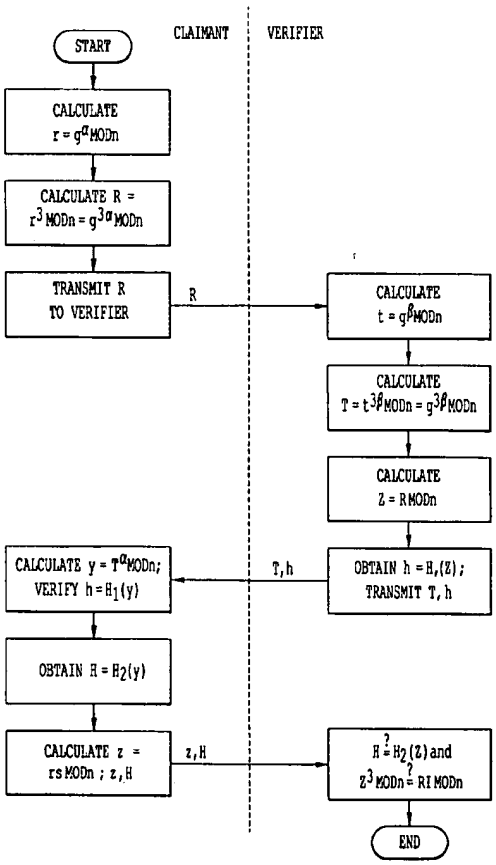
0 311 470 4/1989 European Pat. Off. .

Primary Examiner—Tod R. Swann
Assistant Examiner—Steve Kabakoff
Attorney, Agent, or Firm—Oblon, Spivak, McClelland, Maier & Neustadt, P.C.

[57] **ABSTRACT**

A process for the identification of a claimant by a verifier. The process is of the public key type, where the public exponent is equal to 3. The claimant draws at random a first exponent α, calculates $r=g^{\alpha}$ mod n and transmits $R=r^3$. The verifier draws at random a second exponent β, calculates $t=g^{\beta}$ mod n, calculates $T=t^3$ mod n and $h=H_1(Z)$, where H_1 is a hash function, and calculates $Z=R^3$ mod n. The verifier transmits to the claimant the numbers T and h. The claimant calculates $Y=T^{\alpha}$ mod n, verifies the result $H_1(Y)$, calculates $H=H_2(Y)$, where H_2 is another hash function, calculates $z=rS$ mod n, and transmits z and H. The claimant also has a secret number S equal to the modulo n cubic root of a number I deduced from its identity so that the number S verifies $S^3=I$ mod n. The verifier verifies that H received is equal to $H_2(Z)$ and that z^3 is equal to RI mod n.

2 Claims, 3 Drawing Sheets



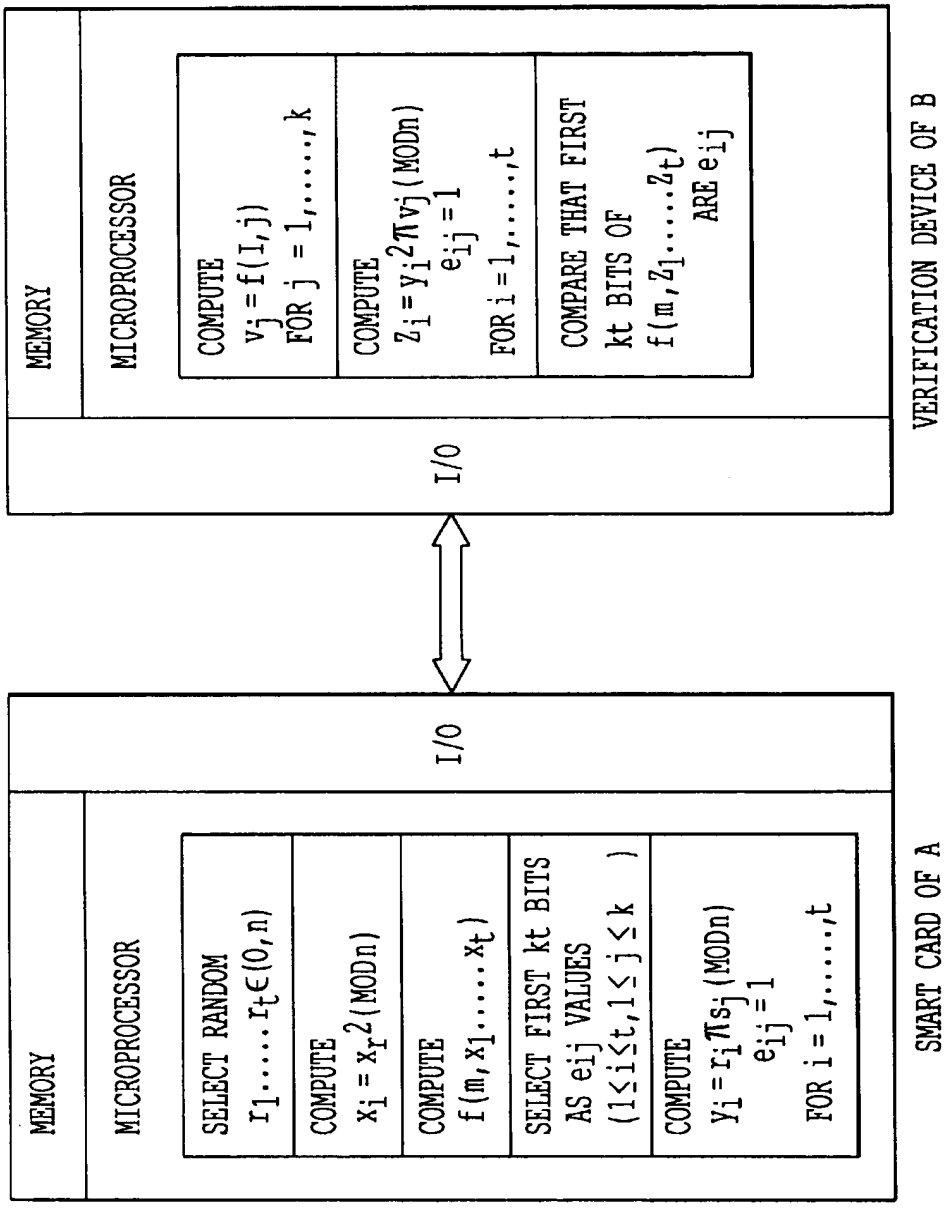


FIG. 1
PRIOR ART

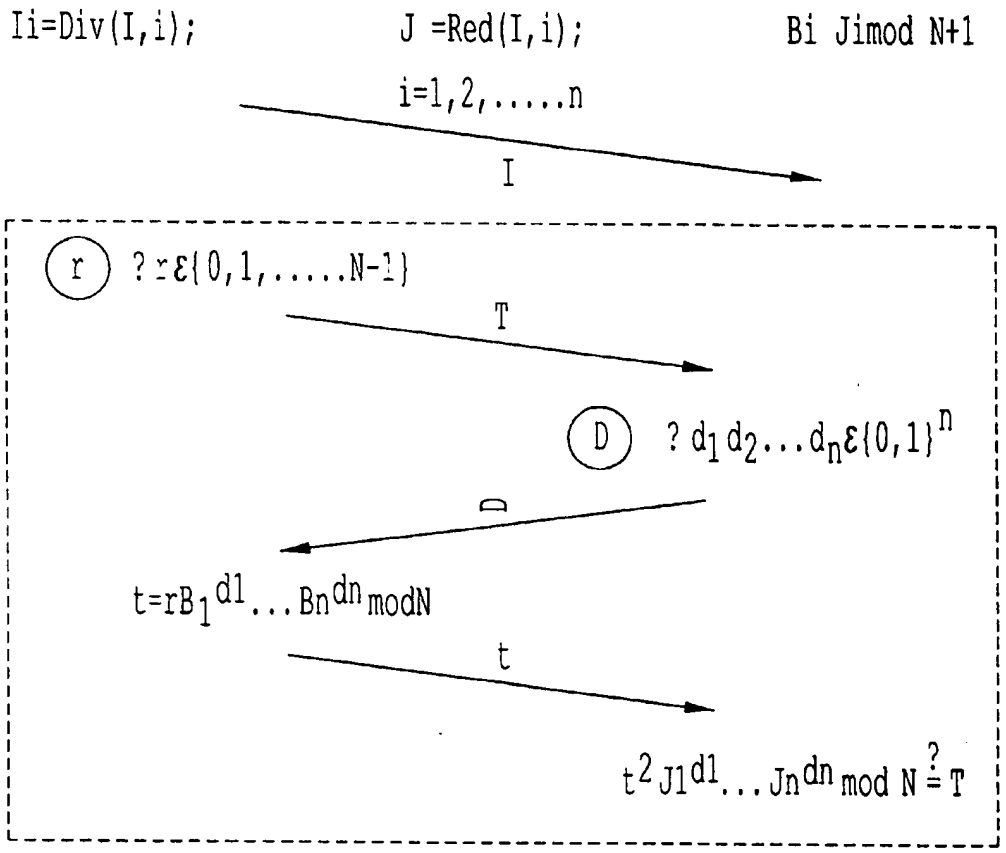


FIG. 2
PRIOR ART

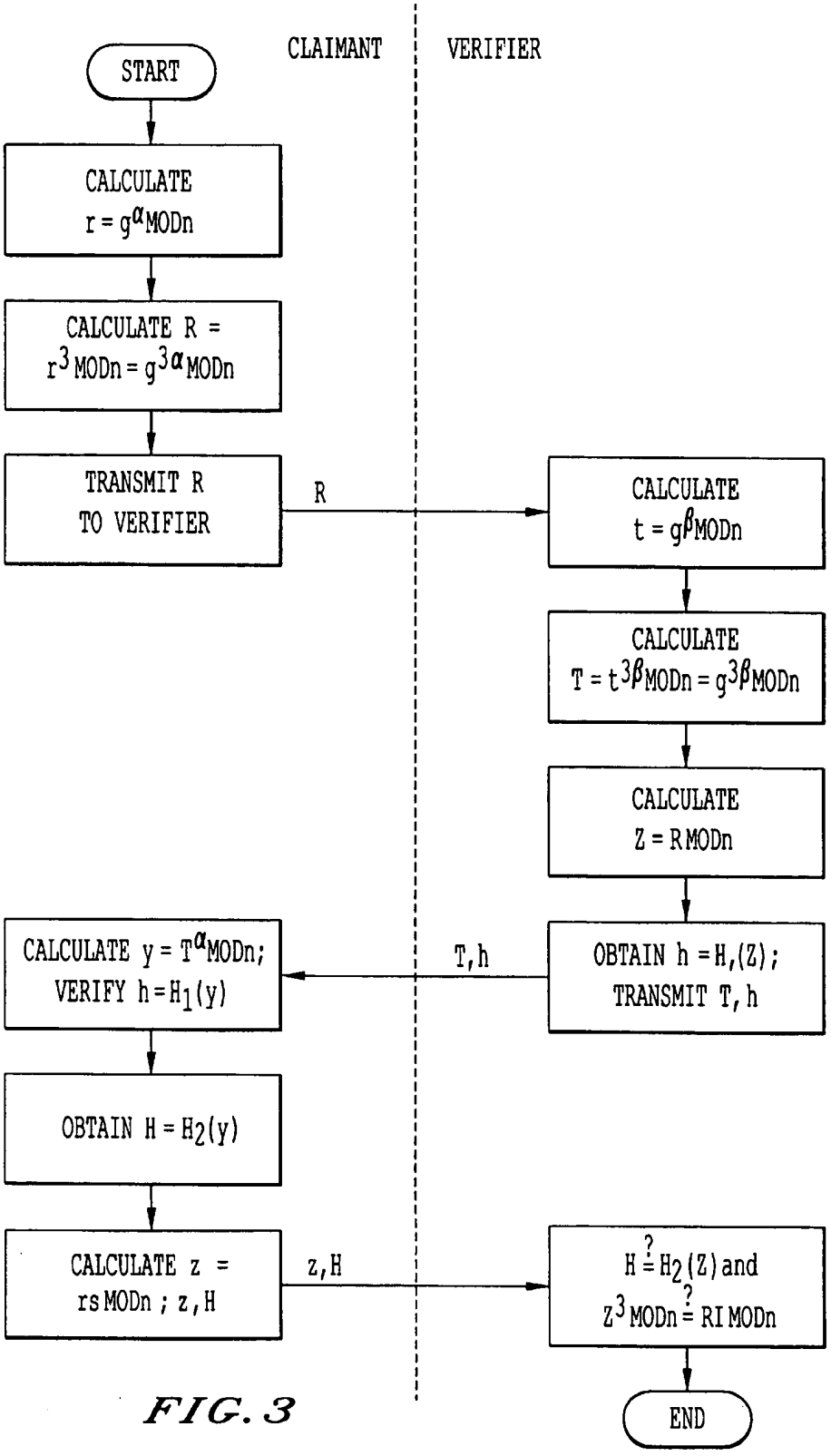

CLAIMANT
VERIFIER
START
CALCULATE $r = g^{\alpha} MODn$
CALCULATE R = $r^3 MODn = g^{3\alpha} MODn$
TRANSMIT R TO VERIFIER
R
CALCULATE $t = g^{\beta} MODn$
CALCULATE $T = t^{3\beta} MODn = g^{3\beta} MODn$
CALCULATE $Z = R MODn$
OBTAIN $h = H_1(Z)$; TRANSMIT T, h
T, h
CALCULATE $y = T^{\alpha} MODn$; VERIFY $h = H_1(y)$
OBTAIN $H = H_2(y)$
CALCULATE z = $rs MODn$; z, H
z, H
$H \stackrel{?}{=} H_2(Z)$ and $Z^3 MODn \stackrel{?}{=} RI MODn$
END

FIG. 3

PUBLIC KEY IDENTIFICATION PROCESS USING TWO HASH FUNCTIONS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a cryptographic identification process enabling a random support, called an identity module (e.g., a smart card, microprocessor, computer, etc.), to prove its identity to means implementing an application, or an interlocutor having verification means, using a protocol setting into action, without revealing the same, one or more secrets contained in the support.

Thus, an identification protocol is a dialogue, through a telecommunications network, between two entities, on the one hand a first entity wishing to prove its identity and which can, if appropriate, be equipped with a terminal (e.g., a computer having a smart card reader) and on the other hand a second entity able to dialogue with the first and perform certain verification calculations.

The first entity, whose identity is to be verified or checked, is hereinafter called the claimant and the second is called the verifier.

The present invention more particularly relates to a public key identification process, where the verifier has no need to know the secrets contained in the identity module of the claimant, but only non-confidential data (the public key) in order to carry out verification calculations.

2. Discussion of the Background

The RSA (initials of the authors RIVEST, SHAMIR, ADLEMAN) public key encryptation algorithm is described in U.S. Pat. No. 4,405,829. At present, it is the most widely used public key algorithm. It supplies signature diagrams also usable for identification purposes.

In the RSA algorithm, a choice is made of two separate prime numbers p and q and their product n is formed. A choice is also made of an integer e, which is prime with the smallest common multiple of (p−1) and (q−1) (or, if desired, which is prime with the product (p−1)(q−1).

In order to encrypt a message, previously placed in digital form u, u being between 0 and n−1, the eth power of u is calculated in the ring of modulo n integers, i.e. $v=u^e$ mod n. It is pointed out that the value of a modulo x integer integer n is equal to the remainder of the division of x by n.

For decrypting a message such as v, it is necessary to extract the eth root of the encrypted message v in the ring of the modulo n integers. This operation amounts to raising the number v to the power d, d being the inverse of the modulo e exponent, the smallest common multiple of the numbers (p−1) and (q−1). If the prime factors p and q are not known, the determination of d is impossible and, with it, the decrypting operation.

One of the first practical uses of the RSA process for identification purposes has been the following: an authority, responsible for the putting into place of an identification system, emits a RSA-type public key, i.e. in practice the two numbers n and e, said key being common to the complete system, and retains the corresponding secret elements (p and q). In each identity module of system users, said authority deposits the pair constituted by:

the identification number ID of the identity module;

the eth rooth (or the inverse of the eth root), modulo n, of a number obtained from the number ID by applying to ID a redundancy function known by everyone (whereof an example can be found in ISO standard 9796), said eth root (or its inverse), calculated by the emission authority with the aid of secret elements held by it, is called "accreditation".

The accreditations deposited in the identity modules can, initially, be used for passive identification purposes (i.e., requiring no calculation on the part of the party wishing to prove its identity). For the verifier, the protocol is then reduced to the following operations:

reading the identity-accreditation pair contained in an identity module;

calculating the eth power of the accreditation and ensuring that the result of this calculation and the application of the redundancy function to the identification number ID do indeed provide the same result.

Such a passive identification demonstrates to the verifier that the party wishing to prove identity has data which can only have been emitted by the authority, which to a certain extent limits identity usurpations. However, nothing prevents a pirate able to intercept the claimant-verifier protocol or a dishonest verifier, from reusing for his own advantage the data supplied by the claimant.

Despite this fraud risk by reuse, the aforementioned passive identification is widely used in the banking field and in the field of telecommunications or phone cards. Supplementary precautions (black lists, etc.) to a certain extent limit the magnitude of frauds by reuse.

However, to solve the problem of fraud by the reuse of exchanged data and which is inherent in passive identification protocols, active identification protocols, i.e. requiring calculations on the part of the party wishing to prove identity, have been proposed. These protocols not only include the use of the RSA algorithm for signing a random question posed by the verifier, but also interactive diagrams where the claimant demonstrates to the verifier that he has one or more accreditations of the type defined hereinbefore and without revealing said accreditation or accreditations. The most widely used of such diagrams are the FIAT-SHAMIR and GUILLOU-QUISQUATER diagrams respectively shown in FIGS. 1 and 2. The FIAT-SHAMIR identification diagram is described in U.S. Pat. No. 4,748,668. The GUILLOU and QUISQUATER identification diagram is described in FR-A-2 620 248 (or its corresponding EP-A-311 470 or corresponding U.S. Pat. No. 5,218,637).

These two diagrams consist of one or more iterations of a base variant with three passes, in which:

1. the party wishing to prove identity (the claimant) calculates the eth power modulo n of a random number r which he draws and deduces therefrom a number x, called the control and which he supplies to the verifier;
2. the verifier draws at random a number b, called the question and sends it to the claimant;
3. the claimant calculates e.g. the product of the random number r by the bth power of his accreditation, i.e. $y=rS^b$ mod n and sends the result y to the verifier, who can calculate the eth power of y and, as he knows the eth power of the accreditation S of the claimant, he is then able to verify consistency between x, b and y.

These diagrams offer a double advantage for active identification. On the one hand, if it is possible to accept an insecurity level (defined as the maximum probability of success of a defrauder) of approximately 10^{-6}, they are much less costly with respect to calculation time than a RSA signature. On the other hand, at least in their basic version are based on zero knowledge disclosure, so that exchanges linked with an identification procedure cannot assist a defrauder in seeking secret accreditations of a user.

Two configurations can be envisaged for implementation, namely on the claimant side, active identification diagrams demonstrating the possession of accreditations of the type described hereinbefore. In a first configuration, the identity module containing the accreditations has an adequate calculation power for performing all the calculations on this side. In a second configuration the identity module containing the accreditations does not perform the calculations itself, but instead allows them to take place in a terminal (e.g., a microcomputer able to read the accreditations in the identity module).

The second configuration, although slightly less reliable than the first, can still be useful for improving the security of the verification of identity modules initially designed for a passive identification. It is necessary to have confidence in the terminal used on the claimant side, but provided that said terminal is integrated, it is not possible for any fraud to come from the network or the verifier.

In the present invention, more particular interest is attached to the problem of use, according to the second configuration, of identity supports initially designed for a passive identification, in which a single accreditation corresponding to a public exponent e equal to 3 has been deposited. Most French bank cards, as well as other identity supports (e.g. telecommunications cards) are of this type.

The GUILLOU-QUISQUATER process is in theory usable by the terminal on the claimant side, for demonstrating to the verifier the possession of the accreditation. In this particular case, the GUILLOU-QUISQUATER process comprises the following operations:

a) two large prime numbers p and q define the integer n, the product of p by q, the number n being rendered public;

b) the calculation support having to prove its identity contains a secret accreditation S between 1 and n−1, the modulo n accreditation cube, i.e. $I=S^3 \bmod n$, being rendered public;

c) the support of the claimant is provided with means able to draw at random an integer r between 1 and n−1 and calculate the cube of r modulo n, called the control x: $x=r^3 \bmod n$;

d) the claimant transmits the control x to the verifier;

e) the verifier draws at random an integer b lower than the exponent 3, i.e. equal to 0, 1 or 2, said integer being called the question;

f) the verifier transmits the question to the claimant;

g) the claimant calculates the number y defined by: $y=rS^b \bmod n$;

h) the claimant transmits the number y to the verifier;

i) the verifier raises to the cube the number y and calculates the product of the control x (which has been transmitted to him) by the power b of I (b drawn by him and I which is public), the verifier then comparing y^3 and $xI^b \bmod n$—if consistency arises, the claimant has correctly replied to the question and his authenticity is assumed.

The security of such a diagram is based on the very hypothesis of the RSA diagram. As both the integer n and the exponent 3 are public, it is difficult for a third party defrauder, to arrive at r by taking the cubic root of x, without knowing the factors p and q, whereof n is the product. Without the knowledge of r, the defrauder cannot correctly reply to the question posed by the verifier.

For such a process, as well as for other hitherto known identification diagrams, the situation where there is only a single accreditation corresponding to a public exponent equal to 3 leads to protocols which are very costly as regards communications. Thus, the security level of a basic exchange (control, question, answer) implementable under the aforementioned conditions is lower than or equal to 3 for the GUILLOU-QUISQUATER diagram. In order to arrive at an appropriate security level (insecurity below 2^{-16}), it is consequently necessary to repeat the basic exchange at least a dozen times, which leads to an increase in the number of bits to be exchanged between the claimant and the verifier by a factor of at least ten.

SUMMARY OF THE INVENTION

The object of the present invention is to obviate this disadvantage. It consists of proposing a diagram, which is both realistic as regards to calculation time and less costly with regards to the number of bits exchanged, making it possible to demonstrate the possession of an accreditation corresponding to a public exponent equal to 3, without revealing it.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the invention and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

FIG. 1 illustrates a Fiat-Shamir diagram;

FIG. 2 illustrates a Guillou-Quisquater diagram; and

FIG. 3 is a flow chart illustrating the operations of the claimant and verifier according to the present invention.

DESCRIPTION OF THE INVENTION

The process according to the invention is based on the following, standard security hypothesis, known as the Diffie and Hellman hypothesis: given an integer n of adequate size, an integer g lower than n, and two integral powers of g modulo n designated $g^{\alpha} \bmod n$ and $g^{\beta} \bmod n$, it is difficult to calculate $g^{\alpha\beta} \bmod n$ without knowing either α or β.

Under this hypothesis, the invention relates to a process for the identification of a support, called "the claimant", by means called "the verifier", said support and said means being equipped with appropriate calculation and storage means, the claimant and verifier having in common:

a first integer n, which is the product of two prime numbers (p, q),

a second integer g between 0 and n−1 and of high order k (the order k being defined as the smaller of the numbers such that $g^k=1 \bmod n$),

a parameter m determining the interval [0, m−1] in which are drawn the random exponents,

a first and second separate hash functions H_1, H_2 and which are independent of one another,

the claimant also having a secret number S equal to the modulo n cubic root of a number I deduced from his identity, so that the number S verifying: $S^3=I \bmod n$, wherein the claimant and verifier utilize their calculation and storage means for performing the following, successive operations:

phase A: the claimant:

Aa) draws at random a first integral exponent α between 0 and m−1,

Ab) calculates a number r equal to the power α of the modulo n number g, namely $r=g^{\alpha} \bmod n$,

Ac) calculates a number R equal to the cube of r modulo n, namely: $R=r^3 \bmod n=g^{3\alpha} \bmod n$,

Ad) transmits the number R to the verifier;

phase B: the verifier:

Ba) draws at random a second integral exponent β between 0 and m−1,

Bb) calculates a number t equal to the power β of the g modulo n number,namely: $t=g^{\beta} \bmod n$,

Bc) calculates a number T equal to the cube of t modulo n, namely: $T=t^3 \bmod n=g^{3\beta} \bmod n$,

Bd) calculates a number Z equal to the power β of R modulo n, namely: $Z=R^{\beta} \bmod n$,

Be) applies the first hash function H_1 to Z for obtaining a number h: $h=H_1(Z)$,

Bf) transmits to the claimant the numbers T and h;

phase C: the claimant:

Ca) calculates a number Y equal to the power α of the number T modulo n, namely: $Y=T^{\alpha} \bmod n$,

Cb) applies to the number Y the first hash function H_1 and obtains the result $H_1(Y)$ and verifies whether this result is equal to the number h received from the verifier,

Cc) applies to the number Y the second hash function H_2 for obtaining a result H: $H=H_2(Y)$,

Cd) calculates a number z equal to the product of the number r by the secret S modulo n, namely: $z=rS \bmod n$,

Ce) transmits the numbers z and H to the verifier;

phase D: the verifier:

Da) applies to the number Z the second hash function H_2 and verifies whether the result obtained $H_2(Z)$ is equal to the number H received from the claimant,

Db) calculates, on the one hand, the product of R by I modulo n and, on the other hand, the cube of z modulo n and checks whether the two results are equal, the identification of the claimant by the verifier being made if the three verifications Cb) Da) Db) are performed.

The parameter m, which the claimant and verifier have in common, can be chosen with the same order of magnitude as k, or equal to n. The value of m must not reveal that of k, which is not known either to the claimant or to the verifier.

The following table summarizes the different operations. Note, FIG. 3 is a flow chart illustrating the same operations.

The horizontal band marked 0 indicates the data known by both entities, namely on the one hand, the number n, number g and cube of the secret I and, on the other hand, the two hash functions H_1 and H_2.

TABLE 1

	Claimant		Verifier
0	n, g, I H_1, H_2 $\alpha < m$		n, g, I H_1, H_2
A	$r = g^{\alpha} \bmod n$		
	$R = r^3 = g^{3\alpha} \cdot \bmod n$	R →	
	$\beta < m$		
			$t = g^{\beta} \bmod n$ $T = t^3 = g^{3\beta} \bmod n$
B			$Z = R^{\beta} \bmod n$
		← T, h	$h = H_1(Z)$
	$Y = T^{\alpha} \bmod n$		
C			

TABLE 1-continued

	Claimant		Verifier
	$h \stackrel{?}{=} H_1(Y)$		
	$H = H_2(Y)$		
	$z = rS \bmod n$	z, H →	
	$H \stackrel{?}{=} H_2(Z)$		
D			$z^3 \stackrel{?}{=} RI \bmod n$

The horizontal band A gives the first operations performed by the claimant (operations Aa to Ad in the above definition).

The horizontal band B gives the following operations performed by the verifier (operations Ba to Bf).

The horizontal band C gives the operations again performed by the claimant (operations Ca to Ce).

Finally, the horizontal band D gives the two final operations performed by the verifier.

Such a process makes it possible to check the authenticity of the holder of an accreditation. Thus, if the claimant knows the secret S, he can correctly reply to the questions asked by the verifier, because he can calculate the quantity $z=rS \bmod n$.

Conversely, in order to be accepted, the claimant must ensure that the equation $H=H_2(Z)$ is satisfied and, for this purpose, after supplying R to the verifier and receiving $T=g^{3\beta} \bmod n$, he must be able to supply a number H, such that $H=H_2(R^{\beta})$:

1. either H is not calculated with the aid of the hash function H_2, then, by admitting that H_2 can be modelled as a random function, there is a negligible probability of relation $H=H_2(R^{\beta})$ is satisfied;
2. or H is the result of H_2 on a value Y, then (unless there is a collision of H_2) Y is equal to Z, in which case, on the basis of $T=g^{3\beta}$, the claimant is able to calculate $T^{\alpha}=g^{3\alpha\beta}$ and then, on the basis of the starting hypothesis, he knows such that $R=g^{3\alpha} \bmod n$.

In addition, the claimant must supply a number z, so that the relation $z^3=RI \bmod n$ is satisfied. For this purpose, he must supply a cubic root of RI, such that $RI=g^{3\alpha}I \bmod n$, then $I=(zg^{-\alpha})^3 \bmod n$.

It is also possible to prove, by means of a few supplementary hypotheses of a not very restrictive nature, that a defrauder using all imaginable fraudulent procedures cannot obtain any information on the accreditation of the claimant and consequently usurp his identity.

Thus, assuming that said defrauder interrogates the claimant (passing himself off as the verifier) with a view to extracting from him an information on the accreditation S:

a) when confronted with an honest claimant, the defrauder is obliged to ask questions honestly and to satisfy the equation $h=H_1(Y)$, after receiving $R=g^{3\alpha}$, he must supply T and h, such that $h=H_1(T^{\alpha})$:
   1. either h is not calculated with the aid of H_1, then, by admitting that H_1 can be modelled as a random function, there is a negligible satisfaction probability;
   2. or h is the result of H_1 on a value Z, then, unless there is a collision of H_1, we obtain Y=Z and in this case, on the basis of $R=g^{3\alpha}$, the verifier is able to calculate

$T^{\beta}=g^{3\alpha\beta}$ and on the basis of the starting hypothesis, he knows β such that $T=g^{3\beta}$ mod n;

b) in the case where the secret S can be expressed as an integral power g^{ν} of g and where the value of m is close to a multiple of k, it is possible to simulate the claimant without knowing the secret S, which means that the interactions which the defrauder can have with the claimant will not enable him to learn anything about S and the simulator will then perform the following operations:

1. he chooses a number δ below m,
2. he calculates $z=g^{\delta}$ mod n, $z^3=g^{3\delta}$ mod n and $R=z^3I^{-1}$, i.e. $g^{3(\delta-\nu)}$ mod n, (the numbers R and z formed in this way roughly follow the same distribution as $R=g^{3\alpha}$ and $z=g^{\alpha+\nu}$),
3. he supplies R to the verifier, who returns T and h—as shown hereinbefore, the verifier is necessarily honest, which means that he knows β such that $T=g^{3\beta}$ mod n and can consequently have knowledge of β (then $T=g^{3\beta}$ mod n, $Z=R^{\beta}$ mod n and $h=H_1(Z)$),
4. he calculates $Y=R^{\beta}=Z$ mod n and $H=H_2(Y)$,
5. he supplies (z,H) to the verifier.

Therefore the above-defined process is reliable and secure, even when confronted with active attacks.

What is claimed is:

1. A process for an identification of a calculation and storage means claimant by a verifier, the claimant and verifier having calculation and storage mechanisms,

wherein the claimant and verifier have the following in common:

a first integer n, which is the product of two prime numbers (p, q),

a second integer g between 0 and n−1 and of high order k, the order k being defined as the smaller of the numbers such that $g^k=1$ mod n,

a parameter m determining the interval in which are drawn the random exponents,

first and second separate hash functions H_1, H_2 and which are independent of one another,

wherein the claimant further includes a secret number S equal to the modulo n cubic root of a number I deduced from its identity, so that the number S verifies: $S^3=I$ mod n,

wherein the claimant and verifier utilize their calculation and storage mechanisms for performing the following, successive operations:

phase A: the claimant:

Aa) draws at random a first integral exponent α between 0 and m−1,

Ab) calculates a number r equal to the power α of the modulo n number g, namely; $r=g^{\alpha}$ mod n,

Ac) calculates a number R equal to the cube of r modulo n, namely: $R=r^3$ mod $n=g^{3\alpha}$ mod n, and

Ad) transmits the number R to the verifier;

phase B: the verifier:

Ba) draws at random a second integral exponent β between 0 and m−1,

Bb) calculates a number t equal to the power β of the g modulo n number, namely: $t=g^{\beta}$ mod n,

Bc) calculates a number T equal to the cube of t modulo n, namely: $T=t^3$ mod $n=g^{3\beta}$ mod n,

Bd) calculates a number Z equal to the power β of R modulo n, namely: $Z=R^{\beta}$ mod n,

Be) applies the first hash function H_1 to Z for obtaining a number h: $h=H_1(Z)$, and

Bf) transmits to the claimant the numbers T and h;

phase C: the claimant:

Ca) calculates a number Y equal to the power α of the number T modulo n, namely: $Y=T^{\alpha}$ mod n,

Cb) applies to the number Y the first hash function H_1 and obtains the result $H_1(Y)$ and verifies whether this result is equal to the number h received from the verifier,

Cc) applies to the number Y the second hash function H_2 for obtaining a result H: $H=H_2(Y)$,

Cd) calculates a number z equal to the product of the number r by the secret S modulo n, namely: z=rS mod n, and

Ce) transmits the numbers z and H to the verifier;

phase D: the verifier:

Da) applies to the number Z the second hash function H_2 and verifies whether the result obtained $H_2(Z)$ is equal to the number H received from the claimant, namely: $H=H_2(Z)$ and Z^3 mod n=RI mod n, and

Db) calculates the product of R by I modulo n and the cube of z modulo n and checks whether the two results are equal, and

wherein the identification of the claimant by the verifier is made if the three verifications determined in steps Cb), Da), and Db) are performed.

2. The process according to claim 1, wherein the claimant is a smart card.

* * * * *

US006134326A

United States Patent [19]

Micali

[11] Patent Number: 6,134,326

[45] Date of Patent: *Oct. 17, 2000

[54] **SIMULTANEOUS ELECTRONIC TRANSACTIONS**

[75] Inventor: **Silvio Micali**, Brookline, Mass.

[73] Assignee: **Bankers Trust Corporation**, New York, N.Y.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

This patent is subject to a terminal disclaimer.

[21] Appl. No.: **08/832,071**

[22] Filed: **Apr. 2, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/751,217, Nov. 18, 1996, Pat. No. 5,666,420.

[51] **Int. Cl.**[7] **H04L 9/00**

[52] **U.S. Cl.** **380/30**

[58] **Field of Search** 380/30, 282, 285; 455/899

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,200,770	4/1980	Hellman et al.	380/30
4,218,582	8/1980	Hellman et al.	380/30
4,405,829	9/1983	Rivest et al.	380/30
4,438,824	3/1984	Mueller-Schloer	380/30
4,458,109	7/1984	Mueller-Schloer	380/30
4,789,928	12/1988	Fujisaki .	
4,885,777	12/1989	Takaragi et al. .	
4,885,789	12/1989	Burger et al. .	
4,953,209	8/1990	Ryder, Sr. et al. .	
5,117,358	5/1992	Winkler .	
5,202,977	4/1993	Pasetes, Jr. et al. .	
5,214,700	5/1993	Pinkas et al. .	
5,220,501	6/1993	Lawlor et al. .	
5,243,515	9/1993	Lee .	
5,276,737	1/1994	Micali	380/30
5,315,658	5/1994	Micali .	
5,440,634	8/1995	Jones et al.	380/30
5,453,601	9/1995	Rosen .	
5,455,407	10/1995	Rosen .	
5,497,421	3/1996	Kaufman et al. .	
5,509,071	4/1996	Petrie, Jr. et al. .	
5,553,145	9/1996	Micali .	
5,610,982	3/1997	Micali	380/30
5,666,420	9/1997	Micali	380/30

OTHER PUBLICATIONS

Abad-Peiro et al., "Designing a Generic Payment Service" (Nov. 26, 1996).

Asokan et al., "Optimistic Protocols for Multi-Party Fair Exchange," IBM Research Report RZ 2892 (Dec. 9, 1996).

Asokan et al., "Optimistic Fair Exchange of Digital Signatures," IBM Research Report.

Asokan et al., "State of the Art in Electronic Payment Systems," IEEE Computer, Sep. 1997, pp. 28–35.

Asokan et al., "Optimistic Fair Exchange of Digital Signatures" Advances in Cryptology (K. Nyberg, ed.), Proc. Eurocrypt 198 , pp. 591–606 (1997).

(List continued on next page.)

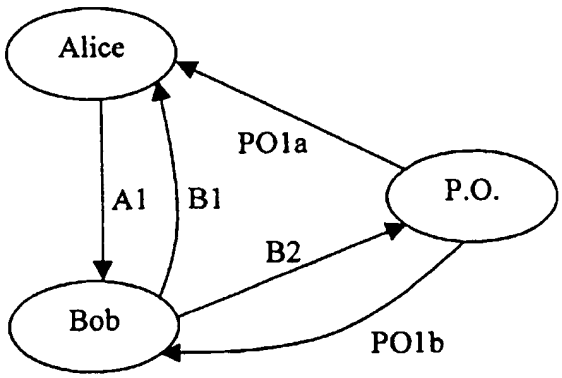
Primary Examiner—Salvatore Cangialosi

Attorney, Agent, or Firm—Pillsbury Madison & Sutro LLP

[57] **ABSTRACT**

A communication method between a first and second party, in the presence of a trusted party, that enables a transaction in which the second party receives a first value produced by the first party and unpredictable to the second party if and only if the first party receives a second value produced by the second party and unpredictable to the first party. The method includes two basic steps: exchanging a first set of communications between the first and second parties without participation of the trusted party to attempt completion of the transaction, and if the transaction is not completed using the first set of communications between the first and second parties, having the trusted party take action to complete the transaction.

75 Claims, 1 Drawing Sheet



OTHER PUBLICATIONS

Asokan et al., "Optimistic Protocols for Fair Exchange," IBM Research Report RZ 2858(Sep. 2, 1996).
Asokan et al., "Server–Supported Signatures," Proceedings of ESORICS '96 (Sep. 25–27, 1996).
Asokan et al., "Server–Supported Signatures," Journal of Computer Security pp. 1–13 (1997).
Asokan et al., "Asynchronous Protocols for Optimistic Fair Exchange," IBM Research Report, Proc. IEEE Symposium on Research in Security and Privacy, pp. 86–99 (1998).
Batelaan et al., "Internet Billing Service Design and Prototype Implementation," Carnegie Mellon University Information Networking Institute 1992 Final Project (Mar. 30, 1993).
Bellare et al., "iKP—A Family of Secure Electronic Payment Protocols" (Jul. 12, 1995).
Ben–Or et al., "A Fair Protocol for Signing Contracts," IEEE Transactions on Information Theory, v. 36 n. 1, pp. 40–46 (Jan. 1990).
Ben–Or et al., "A Fair Protocol for Signing Contrancts," Automata, Languages and Programming, pp. 43–52 (Jul. 1985).
Blum, M., "How to Exchange (Secret) Keys," ACM Transactions on Computer Systems, v. 1 n. 2, pp. 175–193 (May 1983).
Burke et al., "Digital Payment Systems Enabling Security and Unobservability," Computers & Security, v. 8, pp. 399–416 (1989).
Burk et al., "Value Exchange Systems Enabling Security and Unobservability," Computers & Security v. 9, pp. 715–720 (1990).
Camenisch et al., "Digital Payment Systems with Passive Anonymity–Revoking Trustees" Journal of Computer Security (1996).
Camensich et al., "An Efficient Fair Payment System," Proc. 3rd ACM Conf. on Computer Security, pp. 88–94 (1996).
Casey et al., "Secure Automated Document Delivery," Fifth Annual Computer Security Applications Conference, pp. 348–356 (Dec. 4–8, 1989).
Chaum D., "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," Communications of the ACM, v. 24 n. 2, pp. 84–88 (Feb. 1981).
Chaum, D., Security Without Identification: Transaction Systems to Make Big Brother Obsolete, Comm of ACM, vol. 28 No. 10, pp. 1030–1044 (Oct. 1985).
Chaum, et al., Untraceable Electronic Cash, Proc. Crypto '88, pp. 329–327 (1988).
Cheng, et al., Design and Implementation of Modular Key Management Protocol and IP Secure Tunnel ov AIX, IBM Thomas J. Watson Research Center (Apr. 28, 1995).
Chor, et al., Verifable Secret Sharing and Achieving Simultaneity in the Present of Faults, POC, 26th FOCS, pp. 383–395.
Damgard, I., Payment Systems and Credential Mechanisms With Provable Security Against Abuse by Individuals, Proc Crypto '88, pp. 328–335 (1988).
DeMillo, et al., Protocols for Data Security, IEEE Computer, pp. 39–50 (Feb. 1983).
Desmedt, et al., Threshold Cryptosystems, University of Wisconson–Milwaukee, pp. 307–315.
Dolev, et al., Non–Malleable Cryptography, Comm. of ACM, pp. 542–552 (Mar. 1991).
Dukach, S., SNPP: A Simple Network Payment Protocol, M.I.T. Laboratory for Computer Science.
Even, et al., A Randomized Protocol for Signing Contracts, Comm. of the ACM, vol. 28, No. 6, pp. 637–647 (Jun. 1995).
Even, S., Secure Off–Line Electronic Fund Transfer Between Nontrusting Parties, Computer Science Department Technion, Israel Institute of Technology, pp. 1–10 (Jan. 31, 1988).
Even, et al., On–Line/Off–Line Digital Signatures, International Association for Cryptographic Research, 1996, pp. 0–28.
Frankel, et al., Indirect Discourse Proofs: Achieving Efficient Fair Off–Line E–Cash.
Franklin, et al., Fair Exchange with a Semi–Trusted Third Party, Proc. of the 4th ACM Conf. on Computer and Comm. Security, Apr. 1997, pp. 1–6.
Goldreich, et al., How to Play Any Mental Game, Proc. 27th Ann. IEEE ACM Sympoisum on Theory of Computing, pp. 218–229 (1987).
Goldreich, et al., Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero–Knowledge Proof Systems, Association for Computing Machinery, vol. 38, No. 1, pp. 691–729 (Jul. 1991).
Goldwasser, et al., The Knowledge Complexity of Interactive Proof Systems, SIAM J. Comput. vol. 18, No. 1, pp. 186–208 (Feb. 1989).
Herda, S., Constituting evidence and proof in digital cooperation, Computer Standards and Interfaces 17 (1995), pp. 69–79.
Hickman, et al., The SSL Protocol, Netscape Communications Corp. (Jun. 1995).
Jakobsson, M., Reducing Costs in Identification protocols, Crypto '92 (1992).
Janson, et al., Electronic Payment Over Open Networks, IBM Zurich Research Laboratory CH 8803 Ruschlikon, Switzerland (Apr. 18, 1995).
Janson, et al., Electronic Payment Systems, pp. 1–24 (May 1, 1996).
Kilian, J., et al., Identity Escrow, pp. 1–18.
Koleta, G.B., Cryptographers Gather to Discuss Research, Science, pp. 646–647 (Nov. 11, 1981).
Konheim et al., Digital Signatures and Authentications, Cryptography, A Primer, (1981), pp. 334–367.
Low, et al., Anonymous Credit Cards, 2nd ACM Conference on Computer and Communication Security, pp. 1–10 (1994).
Luby, et al., How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically–Biased Coin, IEEE, pp. 11–21 (1983).
Myer, P., Cryptography: A guide for the design and implementation of crytptographic Systems, McGraw–Hill, Inc., pp. 386–430 (1982).
Mueller–Schloer, et al., The Implementation of a Cryptography–Based Secure Office System, AFIPS Conference Proc. 1982, pp. 487–492 (1982).
Needham, et al., Using Encryption for Authentication in Large Netwoks of Computers, Comm. of ACM, vol. 21, No. 12, pp. 993–999 (Dec. 1978).
Pedersen, T., Electronic Payments of Small Amounts, Aarhus Univ. Tech. Rpt. DAIMI PB–495, pp. 1–12 (Aug. 1995).
Otway, et al., Efficient and Timely Mutual Authentication, ACM Operating Systems Review, vol. 21, No. 1, pp. 8–10 (Jan. 1987).
Neuman, et al., Requirements for Network Payment: The NetCheque™ Perspective, Proc. IEEE Compcon '95, San Francisco (Mar. 1995).
Rabin, M., How To Exchange Secrets, (May 20, 1981) pp. 1–21.

Rabin, M., Transaction Protection by Beacons, TR–29–81, Harvard University Center for Research in Computing Technology, (Nov. 1981) pp. 1–21.
Rescorla et al., The Secure HyperText Transfer Protocol, Enterprise Integration Technologies, (Jul. 1995) pp. 1–40.
Rihaczek, K., Teletrust, Computer Networks and ISDN Systems 13 (1987) pp. 235–239.
Rivest et al., A Method for Obtaining Digital Signatures and Public–Key Cryptosystems, Programming Techniques (Feb. 1978) pp. 120–126.
Graham, et al., A Method for Obtaining Digital Signatures and Public–Key Cryptosystems, Programming Techniques, pp. 120–126.
SEMPER Project AC026 Acts Programme (Mar./Jul. 1995).
Serenelli et al., Securing Electronic Mail Systems MILCOM 92 (San Diego, CA 1992) pp. 29.1.1–29.1.4.
Shamir, A., How to Share a Secret, Comm. ACM v. 22, n. 11 (Nov. 1979) pp. 612–613.
Simmons, J., An Impersonation–Proof Identity Verification Scheme, Advances in Cryptology–CRYPTO '87, pp. 211–215.
Simmons, J., Zero–Knowledge Proofs of Identity and Veracity of Transaction Receipts, Advances in Cryptology—EUROCRYPT 188, pp. 35–49.
Simmons, A Protocol to Provide Verifiable Proof of Identity and Unforgeable Transaction Receipts, IEEE Journal on Selected Areas in Communication, vol. 7, No. 4, May 1989, pp. 435–447.
Sirbu, et al., NetBill: An Internet Commerce System Optimized for Network Delivered Services, Engineering and Public Policy Department, Computer Science Department, Carnegie Mellon. University, Pittsburgh, Pennsylvania 15213, pp. 1–11.
Snare, J.L., Secure Electronic Data Interchange, Computer Security in the Age of Information, (W.L. Caelli, ed.), IFIP, 1989, pp. 331–342, (1989).
Sollins, K.R., Cascaded Authentication, IEEE Symposium on Security and Privacy (Apr. 18–21, 1988), pp. 156–163.
Stadler et al, Fair Blind Signatures, Advances in Cryptology—EUROCRYPT '95 (1995).
Stein et al., The Green Commerce Model, pp. 1–17, (Oct. 1994).
Tsudik, G., Zurich iKP Prototype (ZiP), Protocol Specification Document, IBM Zurich Research, pp. i–27 (Mar. 5, 1996).
Varadharajan et al., Formal Specification of A Secure Distributed Messaging System, 12th National Computer Security Conference Proceedings, pp. 146–171, (Oct. 1991).
Varadharajan, V., Notification: A Practical Security Problem in Distributed Systems, 14th National Computer Security Conference, pp. 386–396, (Oct. 1991).
Waidner, M., Development of a Secure Electronic Marketplace for Europe, Proc. of ESORICS 96, Rome, (Seq. 1996), pp. 1–15.
Zhou et al., A Fair Non–Repudiation Protocol, IEEE (1996), pp. 55–61.

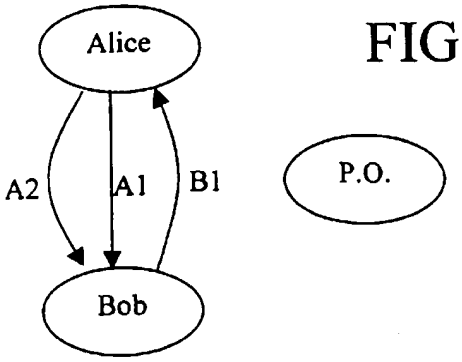


FIG. 1

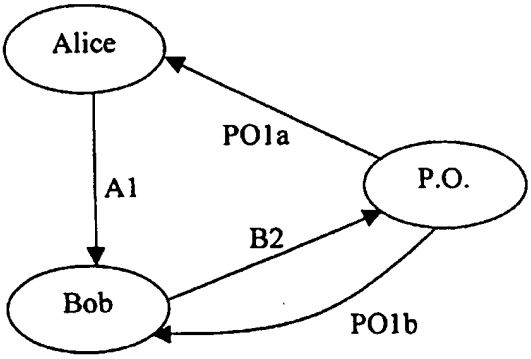


FIG. 2

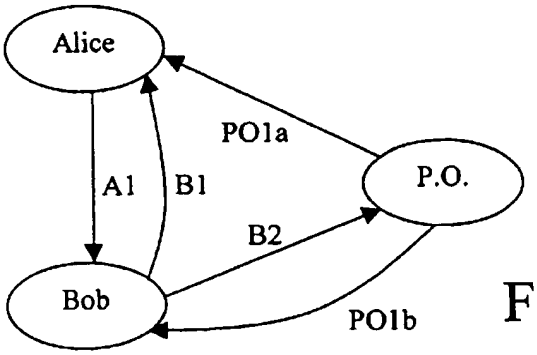


FIG. 3

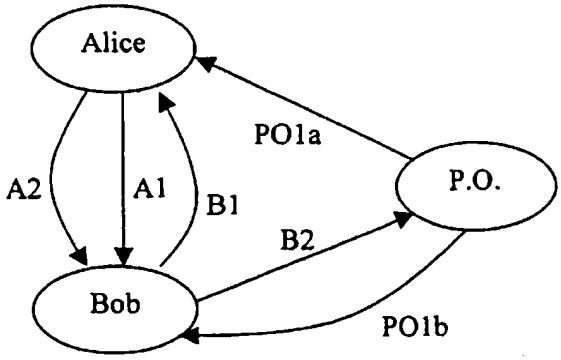


FIG. 4

SIMULTANEOUS ELECTRONIC TRANSACTIONS

This application is a continuation of U.S. patent application Ser. No. 08/751,217, filed Nov. 18, 1996, now U.S. Pat. No. 5,666,420.

TECHNICAL FIELD

The present invention relates generally to electronic commerce and transactions and more particularly to techniques for enabling users to effect certified mail, contract signing and other electronic notarization functions.

BACKGROUND OF THE INVENTION

The value of many transactions depends crucially on their simultaneity. Indeed, simultaneity may be so important to certain financial transactions that entities often are willing to incur great inconvenience and expense to achieve it. For example, consider the situation where two parties have negotiated an important contract that they now intend to "close." Often, the parties find it necessary to sign the document simultaneously, and thus they meet in the same place to watch each other's actions. Another example is the process of certified mail, where ideally the sender of a message desires that the recipient get the message simultaneously with the sender's obtaining a "receipt". A common certified mail procedure requires a person who delivers the mail to personally reach the recipient and obtain a signed acknowledgement when the message is delivered. This acknowledgement is then shipped to the sender. Again, this practice is costly and time consuming. Moreover, such acknowledgements do not indicate the content of the message.

In recent years, the cost, efficiency and convenience of many transactions have been improved tremendously by the availability of electronic networks, such as computer, telephone, fax, broadcasting and others. Yet more recently, digital signatures and public-key encryption have added much needed security to these electronic networks, making such communication channels particularly suitable for financial transactions. Nevertheless, while electronic communications provide speed, they do not address simultaneity.

The absence of simultaneity from electronic transactions severally limits electronic commerce. In particular, heretofore there has been no effective way of building so-called simultaneous electronic transactions ("SET's"). As used herein, a SET is an electronic transaction that is simultaneous at least in a "logically equivalent" way, namely it is guaranteed that certain actions will take place if and only if certain other actions take place. One desirable SET would be certified mail, however, the prior art has not addressed this problem effectively. This can be seen by the following consideration of a hypothetical example, called extended certified mail or "ECM".

In an ECM transaction, there is a sender, Alice, who wishes to deliver a given message to an intended recipient, Bob. This delivery should satisfy three main properties. First, if Bob refuses to receive the message (preferably before learning it), then Alice should not get any receipt. Second, if Bob wishes to receive the message, then he will receive it and Alice will get a receipt for the message. Third, Alice's receipt should not be "generic," but closely related to the message itself. Simultaneity is important in this transaction. For instance, Alice's message could be an electronic payment to Bob, and it is desired that she obtains a simultaneous receipt if possible.

Alice could try to get a receipt from Bob of a message in in the following way. Clearly, sending m to Bob in the clear as her first communication does not work. Should this message be her digital signature of an electronic payment, a malicious Bob may loose any interest in continuing the conversation so as to deprive Alice of her receipt. On the other hand, asking Bob to send first a "blind" receipt may not be acceptable to him.

Another alternative is that Alice first sends Bob an encryption of m. Second, Bob sends Alice his digital signature of this ciphertext as an "intermediate" receipt. Third, Alice sends him the decryption key. Fourth, Bob sends Alice a receipt for this key. Unfortunately, even this transaction is not secure, because Bob, after learning the message when receiving Alice's key, may refuse to send her any receipt. (On the other hand, one cannot consider Bob's signature of the encrypted message as a valid receipt, because Alice may never send him the decryption key.)

These problems do not disappear by simply adding a few more rounds of communication, typically consisting of "acknowledgements". Usually, such additional rounds make it more difficult to see where the lack of simultaneity lies, but they do not solve the problems.

Various cryptographic approaches exist in the literature that attempt to solve similar problems, but they are not satisfactory in many respects. Some of these methods applicable to multi-party scenarios propose use of verifiable secret sharing (see, for example, Chor et al), or multi-party protocols (as envisioned by Goldreich et al) for making simultaneous some specific transactions between parties. Unfortunately, these methods require a plurality of parties, the majority of which are honest. Thus, they do not envision simultaneous transactions involving only two parties. Indeed, if the majority of two parties are honest then both parties are honest, and thus simultaneity would not be a problem. Moreover, even in a multi-party situation, the complexity of these prior art methods and their amount and type of communication (typically, they use several rounds of broadcasting), make them generally impractical.

Sophisticated cryptographic transactions between just two parties have been developed but these also are not simultaneous. Indeed, if just two people send each other strings back and forth, and each one of them expects to compute his own result from this conversation, the first to obtain the desired result may stop all communications, thereby depriving the other of his or her result. Nonetheless, attempts at providing simultaneity for two-party transactions have been made, but by using assumptions or methods that are unsatisfactory in various ways.

For example, Blum describes transactions that include contract signing and extended certified mail and that relies on the two parties having roughly equal computing power or knowledge of algorithms. These assumptions, however, do not always hold and are hard to check or enforce anyway. In addition, others have discovered ways to attack this rather complex method. A similar approach to simultaneity has also been proposed by Even Goldreich and Lempel. In another Blum method for achieving simultaneous certified mail, Alice does not know whether she got a valid receipt. She must go to court to determine this, and this is undesirable as well.

A method of Luby et al allows two parties to exchange the decryption of two given ciphertexts in a special way, namely, for both parties the probability that one has to guess correctly the cleartext of the other is slowly increased towards 100%. This method, however, does not enable the parties to

achieve guaranteed simultaneity if one party learns the cleartext of the other's ciphertext with absolute probability (e.g., by obtaining the decryption key); then he can deny the other a similar success.

For this reasons several researchers have tried to make simultaneous two-party transactions via the help of one or more external entities, often referred to as "centers", "servers" or "trustees", a notion that appears in a variety of cryptographic contexts (see, for instance, Needham and Schroder and Shamir). A method for simultaneous contract signing and other transactions involving one trustee (called a "judge") has been proposed by Ben-Or et al. Their method relies on an external entity only if one party acts dishonestly, but it does not provide guaranteed simultaneity. In that technique, an honest party is not guaranteed to have a signed contract, even with the help of the external entity. Ben-Or et al only guarantee that the probability that one party gets a signed contract while the other does not is small. The smaller this probability, the more the parties must exchange messages back and forth. In still another method, Rabin envisions transactions with the help of external party that is active at all times (even when no transaction is going on), but also this method does not provide guaranteed simultaneity.

The prior art also suggests abstractly that if one could construct a true simultaneous transaction (e.g., extended certified mail), then the solution thereto might also be useful for constructing other types of electronic transactions (e.g., contract signing). As noted above, however, the art lacks an adequate teaching of how to construct an adequate simultaneous transaction

There has thus been a long-felt need in the art to overcome these and other problems associated with electronic transactions.

BRIEF SUMMARY OF THE INVENTION

It is an object of the invention to provide true simultaneous electronic transactions.

It is a further object of the invention to provide an electronic transaction having guaranteed simultaneity in a two-party scenario and with minimal reliance and support of a third party.

It is another more specific object of the invention to provide simultaneous electronic transactions between two parties that rely on third parties in a minimal and convenient manner. In particular, it is desired to provide electronic transactions between two parties that guarantee simultaneity via the help of an invisible third party. A third party is said to be "invisible" because it does not need not to take any action if the transaction occurs with the parties following certain prescribed instructions. Only if one of the original parties deviates from these instructions may the other invoke the intervention of the up-to-then invisible third party, who then can still guarantee the simultaneity of the transaction even though it has not participated from its inception.

These and other objects are provided in a communication method between a first and second party, in the presence of a trusted party, that enables a transaction in which the second party receives a first value produced by the first party and unpredictable to the second party if and only if the first party receives a second value produced by the second party and unpredictable to the first party. The method includes two basic steps: exchanging a first set of communications between the first and second parties without participation of the trusted party to attempt completion of the transaction, and if the transaction is not completed using the first set of communications between the first and second parties, having the trusted party take action to complete the transaction.

Where the first party's value is a message and the second party's value is a receipt, the transaction is a certified transmission of the first party's message. Alternatively, the first party's value represents a commitment to a contract and the second party's value represents a commitment to the contract, such that the transaction is a contract closing.

Preferably, according to the method the first party can prove that some information it receives is the second value, and the second party can prove that some information it receives is the first value.

According to the more specific aspects of the method, at least one of the first and second parties and the trusted party can encrypt messages, and at least one of the first and second parties and the trusted party can decrypt messages. The first set of communications includes at least one communication of the first party to the second party of a data string generated by a process including encrypting a second data string with an encryption key of the trusted party. The second data string includes a ciphertext generated with an encryption key of one of the parties, as well as information specifying or identifying at least one of the parties. The first set of communications also includes at least one communication of the second party of a data string generated by a process that includes having the second party digitally sign a data string computed from information received from the first party in a prior communication, wherein the data string generated by the second party is the second party's value.

According to further aspects of the method, if the second party does not get the first value in the first set of communications, the second party sends the trusted party, for further processing, a data string that includes at least part of the data received from the first party. The further processing by the trusted party includes decrypting a ciphertext with a secret decryption key. The trusted party then sends the first party information that enables the first party to compute the second value, and the trusted party sends the second party information that enables the second party to compute the first value. In either case, the trusted party also verifies identity information of at least one of the parties but preferably does not learn the first value.

DETAILED DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference should be made to the following Detailed Description in conjunction with the accompanying drawings in which:

FIG. 1 illustrates a preferred embodiment of the present invention in which a transaction between a sender and a recipient is completed without involvement of a trusted party;

FIG. 2 shows the case where the recipient proceeds directly to the trusted party for resolution without providing a receipt for the message to the sender;

FIG. 3 shows the case where the sender does not provide the recipient with a form of the message readable to the recipient after receiving the message receipt; and

FIG. 4 shows the case where the sender does not provide the recipient with a readable form of the same message included in its original transmission.

DETAILED DESCRIPTION

In each of the schemes described below, there is a user Alice and a user Bob. The "invisible" third party may be a

financial center that facilitates SETs among its customers, including Alice and Bob. For convenience, the following description shows how to make extended certified mail "simultaneous", although the invention is not so limited. In the context of an ECM system, the third party is called the Post Office. As will be seen, however, contrary to ordinary certified mail, the Post Office here is invisible. The inventive scheme is also preferable to ordinary certified mail because the message receipt also guarantees the content of the message. Also, the electronic transaction is faster, more informative and more convenient than traditional certified mail, and its cost should be substantially lower.

In the preferred embodiment, an extended certified mail system is provided using a single "invisible" trustee or "trusted" party. The system is implemented in a computer network, although it should be realized that telephone, fax, broadcast or other communication networks may be used. Thus, without limitation, it is assumed that each user in the system has a computer capable of sending and receiving messages to and from other computers via proper communication channels.

Each user in the system has a unique identifier. Alice's identifier is denoted by A, and Bob's identifier is B. The identifier of the Post Office is denoted by PO. Users and the Post Office can digitally sign messages. Thus, each has a secret signing key and a matching public verification key. If m is a message (string), then $SIG_A(m)$ indicates Alice's signature of m. (It is assumed, for convenience, that m is always retrievable from its signature. This is the case for most signature schemes, and it is otherwise possible to consider a signed message as the pair consisting of the message and its signature.)

Users and the Post Office can encrypt messages by means of a public-key encryption algorithm (e.g., RSA). Thus, each has a public encryption key and a corresponding secret decryption key. $E_A(m)$, $E_B(m)$, and $E_{PO}(m)$ denote, respectively, the encryption of a message m with the public key of Alice, Bob, and the Post Office. For simplicity, it is assumed that these schemes are secure in the sense that each of E_A, E_B, and E_{PO} appear to behave as a random function. The system can be suitably modified if these functions are much less secure.

Again, for simplicity these encryption algorithms are deterministic and uniquely decodable. Thus, given a value y and a message m, all can verify whether y is the encryption of m with, for example, the Post Office's key, by checking whether $E_{PO}(m)$ equals y. (If the encryption scheme is probabilistic, then one may convince another that a string y is an encryption of a message by providing m together with the random bits that were used to encrypt m.) If y is a ciphertext generated by means of the encryption algorithm E, $E^{-1}(y)$ denotes the corresponding cleartext, whether or not E defines a permutation. (It may also be possible to use encryption algorithms that are not uniquely decodable, for instance, if it is hard to decrypt a given ciphertext in two different ways.) For simplicity, messages are encrypted directly with a public-key algorithm, however, one could first encrypt a message conventionally with some key k, and then encrypt k with a public-key algorithm. (Thus, to decrypt m, one need only just decrypt k).

In one preferred embodiment shown in FIGS. 1–4 and outlined below, the ECM method requires 5 possible steps of communication: A1 and A2 for user Alice, B1 and B2 for user Bob, and PO for the Post Office. However, at most 3 steps should have to be executed. If Alice and Bob are both honest (the case shown in FIG. 1), only steps A1, B1, and A2 will be executed, and in this order. Step B2 will be executed only if Alice fails to execute Step A2 properly (the case shown in FIGS. 2 and 3). The execution of Step B2 causes the Post Office to execute its only step, PO (for clarity, shown as two steps, PO1a and PO1b in the Figures). The protocol is as follows:

A1.	Given her message m, Alice computes $z = E_{PO}((A, B, E_B(m)))$, the encryption in the Post Office public key of a triplet consisting of identifiers A, B and the message encrypted in Bob's key, and then sends z to Bob.
B1.	Upon receiving z from Alice, Bob digitally signs it and sends it to Alice as the receipt.
A2.	If Alice receives the properly signed receipt from Bob, she sends m to Bob.
B2.	If, within a given interval of time after having executed Step B1, Bob receives a string m such that $E_{PO}((A, B, E_B(m))) = z$, the value originally received from Alice, then he outputs m as the message and halts (the case of FIG. 1). Otherwise, Bob sends the value z signed by him to the Post Office indicating that Alice is the sender and he is the recipient.
PO.	If Bob's signature relative to z is correct, the Post Office decrypts z with its secret key. If the result is a triplet consisting of A, B and a string x, the Post Office (a) sends Alice the value z signed by Bob as the receipt, and (b) sends x to Bob.

Preferably, in step A1 Alice sends z to Bob digitally signed by her. In addition, in step A1 Alice may sign z in a standard format that indicates z is part of an extended certified mail sent from Alice to Bob, e.g., she may sign the tuple (ECM, A, B, z). In this way, Bob is certain that z comes from Alice and that, when Alice holds a receipt for m signed by Bob in step B1, he will have a certified version of m. Further, if z is digitally signed by Alice in step A1, Bob first checks Alice's signature, and then countersigns z himself in step B1. The adoption of a standard format also insures that, by signing z in step B1 as part of an ECM system, Bob does not sign accidently a message that has been prepared by Alice maliciously. Also, the Post Office may also check Alice's signature or any additional formats if these are used.

In analyzing the protocol, it should be noted that Alice, given Bob's signature of z as receipt in step B1, can prove the content of the message by releasing m. Indeed, all can compute $x = E_B(m)$ and then verify that $E_{PO}((A, B, x))=z$.

Notice also that the Post Office does not understand the message sent in step B2 via the ECM protocol, whether or not it is called into action. Rather, the Post Office can only obtain $E_B(m)$, but never m in the clear (in this embodiment).

Third, notice that m is, by definition, equal to $E^{-1}{}_B(x)$, where $(A, B, x)=E_{PO}{}^{-1}(z)$, and may be non-sensical. Indeed, nothing prevents Alice from sending Bob a garbled message in step A1. However, she can only get a receipt for this same garbled message in step B1. It is also noted that, if not every string is an encryption of some message, Alice may choose z so that it is not the encryption of anything. In such case, however, she cannot ever claim to have a receipt for any message. Alternatively, it may be desirable to use cryptosystems for which either every string is an encryption of some other string or such that it can be easily detected whether y encrypts something.

The protocol works for the following reasons. When receiving the value $z=E_{PO}((A, B, E_B(m)))$ from Alice in step A1, Bob will have difficulty in computing $E_B(m)$, and thus m, from z without the Post Office's secret key. Thus, if he halts, Alice would not get her receipt, but Bob would not get m either.

Assume now that Bob signs z and sends it to Alice (step B1). Because this gives Alice a valid receipt from Bob for

her message m, for the simultaneity constraint to hold, it must be shown that Bob easily obtains m. This is certainly true if Alice sends m to Bob in Step A1. Assume therefore that Alice does not send him m as shown in FIG. 4. Then, Bob presents z signed by him to the Post Office, essentially asking the Post Office to retrieve (for him) $E_B(m)$ from z as shown in FIG. 3. The Post Office complies with this request. In doing so (step PO1*b*), however, the Post Office also sends Alice z signed by Bob as the receipt (step PO1*a*). It does so to prevent one last possibility; that Bob, upon receiving z from Alice in Step A1, rather than sending her the receipt in Step B1, goes directly to the Post Office in order to have $E_B(m)$ extracted from z. This is the case shown in FIG. 2.

Summarizing, if Alice sends a message encrypted with the Post Office key to Bob, and Bob does not send Alice a receipt, or if he does not access the Post Office, Bob will never learn m. Otherwise, Alice is guaranteed to get her receipt for m either from Bob or from the Post Office. On the other hand, upon receiving an encrypted message, Bob is guaranteed that he will understand it, either helped by Alice or helped by the Post Office.

In the preferred embodiment above, the triplet (which includes the ciphertext $E_B(m)$) also includes A and B. The ciphertext is customized in this way so that it can be used by the system only for the purpose of Alice sending a message to Bob. Whether or not this customization is performed, the system is very convenient to use because everyone knows the public key of the Post Office, because everyone can encrypt a value with that key, and because the Post Office can remove this encryption layer for those recipients who claim to have been betrayed by their senders. However, without the above (or an equivalent) customization, this same convenience could be exploited by a malicious recipient, who could learn his messages while denying the senders their legitimate receipts.

In particular, assume that this customization is removed altogether. Then, a malicious Bob, upon receiving $z'=E_{PO}(E_B(m))$—rather than $z=E_{PO}((A, B, E_B(m)))$—from Alice in Step A1, may behave as follows. First, he does not send Alice any receipt. Second, he signs z'. Third, he gives this signed value to the Post Office complaining that a sender Chris (an accomplice of his) is refusing to send him the message in the clear. At this point, the Post Office, after verifying Bob's signature and not having any way of checking whether Chris is the real sender, retrieves $E_B(m)$ from z' and sends $E_B(m)$ to Bob, while simultaneously sending the signed z' to Chris as his receipt. Of course, Chris may destroy or hide this receipt. Meanwhile Alice, who does not get any receipt after Step A1, may think that Bob is away or does not want to receive her message. But she believes that Bob will never be able to read her message in any case.

This violation of the simultaneity constraint (i.e., Bob receiving m while Alice having no receipt) may still occur if, without any customization, Alice signs z when sending it to Bob in Step A1. Indeed, Bob would have no trouble in removing Alice's signature, asking Chris to sign z' and then presenting to the Post Office z' signed by Chris and countersigned by himself. The Post Office, after verifying Bob's and Chris's signatures, would still (after removing its encryption layer) send $E_B(m)$ to Bob and the receipt to Chris. This violation of simultaneity, however, does not occur with the customization of the triplet to include A and B. Indeed, assume that Bob gives the Post Office the value $z=E_{PO}((A, B, E_B(m)))$ originally received by Alice and signed by him and Chris, claiming that it was sent to him by Chris. Then, the Post Office, after verifying Bob's (and Chris's) signature and after computing the value $E_{PO}^{-1}(z)$, will notice that this value—i.e. $(A, B, E_B(m))$—does not specify Chris to be the sender and Bob the receiver.

The benefits of this customization may be implemented in varying ways. For instance, Alice's signature of $(B,E_B(m))$ may be sufficient to indicate that the sender is Alice and the receiver is Bob. More generally, any customization that prevents Bob from obtaining $E_B(m)$ from the Post Office while convincing the Post Office not to send Alice the receipt is within the scope of the invention.

It should be realized that any customization for the purpose of simultaneous electronic transactions is itself within the scope of the present invention, whether or not implemented with an invisible third party. For instance, Alice may send $E_{PO}(A,B,E_B(m))$ directly to the Post Office, which gives $E_B(m)$ to Bob (if Bob signs the receipt for Alice) after checking that Alice and Bob are, respectively, the sender and the receiver. Alternatively, Alice may send the Post Office $E_{PO}(SIG_A(B,E_B(m)))$ for identifying the sender and the recipient in a way that cannot be decoupled from the transaction. Such approaches may be especially useful with a plurality of trustees as described below. Such an approach, which calls into action the trusted party directly with a proper customization step as described, is also useful for hiding the identity of the sender from the recipient. Indeed, the Post Office may solicit a proper receipt from Bob without disclosing Alice's identity (even if the receipt indicates the content of Alice's message).

Although not specified above explicitly, it should be appreciated that all or part of the actions required by the Post Office, Alice or Bob can be realized in software. Some of these actions can also be performed by hardware, or physically secure devices (i.e. devices such as secure chips having at least some portion of which is tamper-proof).

Many variations of the disclosed protocol can be envisioned and are within the scope of the present invention. For instance, while the "receipt" described above witnesses the content of the message sent, the receipt can be made generic, e.g., by having Bob sign a "declaration" (instead of a string including an encrypted version of the message) that he has received an encrypted message from Alice at a given time. Also, if desired, the customization step (i.e. the inclusion of the identifiers A and B in the triplet) can be omitted. This might be advantageous, for example, when no other user may collude with either Alice or Bob to disrupt simultaneity. This may occur where there is no third user, as in the case when certified mail occurs between two predetermined people. In the disclosed system, the Post Office cannot learn the content of the message, but such a restriction can be removed also (e.g., by having Alice compute $z=E_{PO}(A, B, m)$). It may also be convenient to one-way hash strings prior to signing them.

Still another variation would be to impose some temporal element on the transaction. For instance, when Alice sends Bob $z=E_{PO}(A, B, E_B(m))$, she may sign z together with some additional information that specifies a certain time (either absolute or relative to the sending time) after which the Post Office will not help Bob obtain the message. Preferably, Alice specifies this time in a signed manner both outside the Post Office encryption layer as well as within the triplet. In such case, the Post Office must obtain from Bob all necessary information to verify that the time specified outside the PO encryption layer checks with the time specified within the triplet. If it does not, then several possibilities may occur. For example, the Post Office will not help Bob recover the message, or the message is considered unsent (even if Alice obtains a receipt).

Other variations are also possible. Some variations may be used in conjunction or in alternative to the techniques described above. One group of such variants concerns the encryption method used.

For instance, E_B does not need to be interpreted as an encryption algorithm for which Bob has the decryption key. It may just be an encryption algorithm for which Bob can have the message decrypted. For example, and without limitation, the decryption key of E_B may lie with a group of people, each having a piece of the key. These same alternative interpretations apply also to E_A or E_{PO}.

Also, while public-key cryptosystems are quite convenient, it should be realized that conventional cryptosystems could be used for the ECM protocol. For example, x may be the conventional encryption of (A, B, E_B(m)) with a secret key k shared between Alice and the Post Office. This key k may be released if it is desired that Bob verify m to be the genuine message. If, however, it is feared that release of a different key may change the content of the messages, special redundancies could be used. For instance, conventionally a message M is encrypted by actually encrypting (M, H(M)), where R is a one-way function. Thus, if e is an encryption of (M, H(M)) with a key k, it is hard to find a second key K such that e also is an encryption with that key of (M'H(M')). It is preferable that k, rather than being a secret key shared by Alice and the Post Office, is a temporary key that Alice may transfer to the Post Office separately by means of a different shared key K. This way, divulging k (e.g., for the purpose of convincing Bob of the value of E_B(m)) does not force the Post Office and Alice to agree on another conventional key k.

It should also be appreciated that the digital signatures of the ECM system need not be public key signatures. For instance, there may be private key digital signatures or signatures verifiable with the help of other parties, or other suitable forms of message authentication. Thus, as used herein, "digital signatures" and "digital signing" should be broadly construed. Similarly, the notion of encryption with a key of some party should be broadly construed to include encrypting with a public key of that party or encrypting with a secret key shared with that party or known to that party.

There may also be concern that the Post Office will collude with one of the parties. For instance, the Post Office may collude with Bob who, rather than sending the receipt to Alice, goes directly to Post Office, and this enables Bob to understand his message but without giving Alice any receipt. This may occur in ordinary certified mail. Indeed, one who delivers the post may leave a letter with his intended recipient without asking him or her to sign a receipt. Nonetheless, this potential problem may be dealt with effectively and efficiently. For instance, the Post Office may be (or make use of) a physically secure device. Assuming that the Post Office uses such a device in the preferred embodiment, then it will be hard for user Bob to have the Post Office decrypt (A, B, E_B(m)) for him without sending Alice her receipt. Indeed, the chip can be programmed to perform both operations or none. Although use of physically secure devices might increase the cost of a system, this need not be the case. Indeed, while they may be millions of users, there may be one or much fewer Post Offices. (Each user, of course, may benefit also from being or relying upon physically secure devices.)

While the inventive ECM system is very economical as it requires at most three communication steps, the goals can be accomplished also by more steps. In particular, although the trusted party, upon receiving Bob's communication, can enable Bob to get his message and Alice to get her receipt, without sending messages back and forth, this goal can be accomplished by means of a more complex dialogue. Indeed, more elaborate dialogues, and in particular zero knowledge proofs (see, e.g., Goldwasser et al or Goldreich et al) could be useful (also as an alternative to physically secure devices) to give Bob the message or Alice the receipt so that they learn their respective values, but are not able to "prove" these values to third parties.

A further alternative method envisions a Post Office with a plurality of trustees. A multiplicity of trustees can be beneficial for various aspects, particularly, if the system is set up so that more than one of the trustees must collude for cheating to occur. Presumably, however, each trustee is selected with trustworthiness (or, if it is a device, proper functioning) as a criterion, and thus the possibility that more than one of them is malicious or defective is very small.

A simultaneous ECM system with a multiplicity of trustees may make novel use of prior techniques such as fair cryptography, or secret sharing, verifiable secret sharing or threshold cryptosystems.

In a construction based on fair public cryptosystems, the triplets (A, B, E_B(m)) are not encrypted with the Post Office's public key, but rather with a user public key. In this embodiment, user Alice computes a pair of public and secret key of a fair public-key cryptosystem, properly shares her secret key with the trustees of the Post Office (e.g., receiving from said trustees a certification that they got legitimate shares of this user key) in some initial phase, and then performs Step A1 of the above ECM protocol. If needed, Bob may turn to the Post Office and instructs the trustees to reconstruct Alice's user key. By doing so, the trustees cannot monitor or cause the Post Office to monitor the message addressed by Alice to Bob, but can reconstruct the triplet (A, B, E(m)). To insure that the Post Office trustees do not collude with Bob in depriving Alice of her receipt, it can be arranged that each trustee, when contributing its own piece of a user secret key, also gives a proper acknowledgement to that user. Thus, unless all n trustees do not behave properly, Alice would receive at least one receipt.

A possible drawback of this fair-cryptography based system is that Alice must interact with the trustees in order to give them shares of her user key. Thus, the trustees are not fully invisible. This interaction may not even be confined to a single initial phase. This is because Alice may not be able to reuse her key after Bob accesses the Post Office and causes its reconstruction. To alleviate this problem, it might be desirable to use physically secure devices and having the trustees reveal their own pieces to such a device, which would then be able to announce (A,B,E_B(m)) without proof.

A better approach uses the ECM protocol, but involves splitting the secret key of the Post Office rather than the secret user keys. Thus, Alice would continue to encrypt (A, B, E_B(m)) with the help of the Post Office public key, whose corresponding secret key is shared among the n trustees but is not known to any single entity (nor has it been prepared by any single entity). Thus, the n trustees must cooperate, under Bob's proper request, in removing the Post Office's encryption layer. However, they do so without reconstructing the Post Office secret key, not even internally to the Post Office. To this end, a threshold cryptosystem may be used). This solution is now illustrated using the well-known Diffie-Hellman public-key cryptosystem.

In the Diffie-Hellman system, there is a prime p and a generator g common to all users. A user X chooses his own secret key x at random between 1 and p−1, and sets his

public key to be g^x mod p. Let y and g^Y mod p, respectively, be the secret and public keys of user Y. Then X and Y essentially share the secret pair key g^{xy} mod p. Indeed, each of X and Y can compute this pair-key by raising the other's public key to his own secret key mod p. On the other hand, without knowledge of x or y, no other user, given the public keys g^x mod p and g^Y mod p and based on any known method, can compute the pair-key g^{xy}. Thus X and y can use this key to secure communications between each other (e.g., by using it as the key of a symmetric cipher).

Let now $T_1, \ldots, T_n$ be the trustees of the Post Office. Then, each T_i chooses a secret key xi and a matching public key g^{xi} mod p. Then the public key of the Post Office is set to be the product of these public keys mod p, g^z mod p (i.e., $g^z=g^{x1+\ldots+xn}$ mod p). Thus, each trustee has a share of the corresponding secret key, z. Indeed, the Post Office's secret key would be z=x1+. . . +xn mod p−1. Assume now that Alice wishes to encrypt (A, B, E_B(m)) with the Post Office's key. She selects a (preferably) temporary secret key a and its corresponding public key g^a mod p. She then computes the public pair-key g^{az} mod p, encrypts (A, B, E_B(m)) conventionally with the secret pair-key g^{az}, and then sends Bob this ciphertext together with the temporary public-key g^a mod p (all in Step A1). If in Step B1 Bob sends Alice back a receipt, namely, his signature of the received message, then Alice, in Step A2, sends him the secret key a. This enables Bob to compute the pair-key g^{az} mod p (from a and the Post Office's public key), and thus decrypt the conventional ciphertext to obtain (A, B, E_B(m)). Thus, if both users behave properly, the Post Office is not involved in the transaction. Assume now that Bob properly asks the Post Office to decrypt Alice's ciphertext. To do this, the trustees cooperate (preferably, with proper notice to Alice and to each other) in computing g^{az} mod p. To this end, each trustee T_i raises Alice's public key g^a mod p to its own secret key. That is, T_i computes g^{axi} mod p. Then these shares of the pair-key are multiplied together mod p to obtain the desired private pair-key. In fact, $g^{ax1} \ldots g^{axn}$ mod $p=g^{ax1+\ldots+axn}$ mod $p=g^{a(x1+\ldots+xn)}$ mod $p=g^{az}$ mod p. This key may be given to Bob, who can thus obtain E_B(m). In this method, it may be useful to have a Post Office representative handle the communications with Bob, while the individual trustees handle directly their sending Alice receipts.

This method can be adjusted so that sufficiently few (alternatively, certain groups of) trustees cannot remove the Post Office's encryption layer, while sufficiently many (alternatively, certain other groups of) trustees can. For instance, there can be kn trustees, and each of the n trustees acting as above can give his own secret key to each of a group of k−1 other trustees. Thus, each distinct group of k trustees has knowledge of a secret key as above. Further, the above-described modifications to the single invisible-trustee ECM protocol can be applied to embodiments involving multiple trustees.

In the ECM system involving fair cryptography, even a user might be or rely upon a multiplicity of entities. Indeed, in the invention, "user" or "party" or "trusted party" thus should be construed broadly to include this possibility.

It should be appreciated that the inventive ECM systems enable Alice and Bob to exchange simultaneously two special values, the first, produced by Alice, which is (at least reasonably) unpredictable to Bob, and the second, produced by Bob, which is unpredictable to Alice. Indeed, the value produced by Bob and unpredictable to Alice may be Bob's signature of step B1. If the message is not known precisely by Bob, then the message itself may be the value produced by Alice and unpredictable to Bob. Alternatively, if Bob knows the message precisely (but it is desired that he receive it from Alice in an official and certified manner), then the parties may use a customization step so that, for example $SIG_A(m,E_B(m))$ is the value produced by Alice and unpredictable to Bob.

The inventive system is useful to facilitate other electronic transactions that require the simultaneous exchange of unpredictable values. One such example, not meant to be limiting, involves a contract "closing" wherein a pair of users desire to sign a contract at a particular time and place. The invention thus allows Alice and Bob to sign a contract simultaneously with an invisible third party. Indeed, the first value may be Alice's signature of the contract C and the second value Bob's receipt for a message consisting of Alice's signature of C.

In particular, assume that Alice and Bob have already negotiated a contract C. Then, Alice and Bob agree (in a preliminary agreement) (a) that Alice is committed to C if Bob gets the message consisting of Alice's signature to C, and (b) that Bob is committed to C if Alice gets Bob's receipt of that message. This preliminary agreement can be "sealed" in many ways, for instance by signing, preferably standardized, statements to this effect conventionally or digitally. It does not matter who signs this preliminary agreement first because Bob does not have Alice's message and Alice does not have Bob's receipt. However, after both parties are committed to the preliminary agreement, the inventive ECM system allows the message and the receipt to be exchanged simultaneously, and thus C is signed simultaneously. Those skilled in the art also may realize it may be more convenient to first one-way hash C prior to signing it.

This method may be much more practical than accessing a commonly trusted lawyer particularly, when the contract in question may be very elementary or arises in an "automatic context". Generalizing, one may view a contract C as any arbitrary signal or string of symbols to which the parties wish to commit in a simultaneous way. The inventive solution is very attractive because it can be implemented in software in many contexts, and because the trustee is invisible and need not be called into use if the signatories behave properly. This minimizes cost and time, among other resources. In this application, the trustee, rather than a post office, may be a "financial service center" that facilitates the transactions of its own customers.

Yet another application of the invention is to make simultaneous the result of applying a given function to one or more secret values, some belonging to Alice and some belonging to Bob. For example, the inventive method allows implementation of "blind" negotiations. In this embodiment, assume a seller Alice and a buyer Bob desire to determine whether Alice's (secret) minimum selling price is lower than Bob's (secret) maximum selling price (in a way that both parties will learn the result simultaneously). If the answer is no, then the parties may either try again or terminate the negotiation. Alternatively, if the answer is yes, then preferably the parties also will be committed to the transaction at some value. (For example, the average of the two secret values).

Another useful application of the invention is during a bid process, such as in an auction. For instance, assume that multiple bidders wish that their secret bids be revealed simultaneously. One bidder may also wish that his or her bid be independent of the other bids.

What is claimed is:

1. A method of communicating between a first and second party, comprising the steps of:

initiating an exchange of messages between the first and second parties without intervention of a trusted third party; and

in response to one of the first and second parties not receiving at least one of the messages from an other one of the first and second parties, having the trusted third party take action to provide appropriate messages to the first and second parties.

2. An electronic communication method comprising:

sending from a first party a first message to be received by a second party without intervention by a trusted third party, at least some of the contents of the first message being unintelligible to the second party;

receiving by the first party a second message from the second party verifying that the second party received the first message; and

sending from the first party a third message enabling the second party to understand the contents of the first message unintelligible to the second party;

wherein the at least some of the contents are able to be rendered intelligible to the second party through assistance of the first party; and

the at least some of the contents are able to be rendered intelligible to the second party through assistance of the trusted third party.

3. The electronic communication method of claim 2, further comprising generating, by the first party, the first message using information not known by the second party.

4. The electronic communication method of claim 2, wherein the first message includes a valid digital signature by the first party of at least some of the contents unintelligible to the second party.

5. The electronic communication method of claim 2, wherein the first message includes information indicating the first party is involved in the message.

6. The electronic communication method of claim 2, wherein the first message includes information indicating the second party is involved in the message.

7. The electronic communication method of claim 2, wherein the second message has a portion unpredictable to the first party.

8. The electronic communication method of claim 2, wherein the second message includes a valid signature of the second party of information representative of the first message.

9. An electronic communication method comprising:

sending from a first party a first message to be received by a second party without intervention by a trusted third party, at least some of the contents of the first message being unintelligible to the second party; and

receiving by the first party a second message from the trusted third party, the second message verifying that the second party received the first message;

wherein the at least some of the contents are able to be rendered intelligible to the second party through assistance of the first party, and

the at least some of the contents are able to be rendered intelligible to the second party through assistance of the trusted third party.

10. The electronic communication method of claim 9, further comprising generating, by the first party, the first message using information not known by the second party.

11. The electronic communication method of claim 9, wherein the first message includes a valid digital signature by the first party of at least some of the contents unintelligible to the second party.

12. The electronic communication method of claim 9, wherein the first message includes information indicating the first party is involved in the message.

13. The electronic communication method of claim 9, wherein the first message includes information indicating the second party is involved in the message.

14. The electronic communication method of claim 9, wherein the second message has a portion unpredictable to the first party.

15. The electronic communication method of claim 9, wherein the second message includes a valid signature of the second party of information representative of the first message.

16. The electronic communication method of claim 9, wherein at least a part of the first message is unintelligible to the trusted third party.

17. The electronic communication method of claim 9, wherein the second message is based on information not known by the trusted third party.

18. The electronic communication method of claim 9, wherein at least a part of the second message is unintelligible to the trusted party.

19. The electronic communication method of claim 9, further comprising receiving, by the first party after sending the first message and before receiving the second message, a third message.

20. The electronic communication method of claim 19, wherein:

the third message verifies receipt of the first message by the second party; and

the first party sends no other messages to the second party before receiving the second message.

21. The electronic communication method of claim 19, wherein the third message does not verify receipt of the first message by the second party.

22. The electronic communication method of claim 21, wherein the third message does not verify receipt of the first message by the second party because it does not include a valid signature by the second party of information representative of the first message.

23. The electronic communication method of claim 22, further comprising determining, by the first party, that a signature in the third message is not a valid signature of the second party.

24. The electronic communication method of claim 22, further comprising determining, by the first party, that a signature in the third message is not a signature of a given portion of the first message.

25. The electronic communication method of claim 24, further comprising sending, by the first party after sending the first message and before receiving the second message, a third message to be received by the second party, where the third message does not satisfy a predetermined criterion.

26. The electronic communication method of claim 25, wherein the predetermined criterion is one of that the third message enables intelligible disclosure of the contents of the first message unpredictable to the second party.

27. The electronic communication method of claim 25, wherein the predetermined criterion is that the third message does not include a valid signature of information representative of the contents of the first message unpredictable to the second party.

28. An electronic communication method comprising:

receiving by a first party a first message from a second party without intervention of a trusted third party, at least some of the contents of the first message being unintelligible to the first party;

sending by the first party a second message verifying that the first party received the first message; and

receiving by the first party a third message from the second party enabling the first party to understand the contents of the first message unintelligible to the first party;

wherein the at least some of the contents are able to be rendered intelligible to the first party through assistance of the second party, and

the at least some of the contents are able to be rendered intelligible to the first party through assistance of the trusted third party.

29. The electronic communication method of claim **28**, wherein the first message includes a valid digital signature by the second party of at least some of the contents unintelligible to the first party.

30. The electronic communication method of claim **28**, wherein the first message includes information indicating the second party is involved in the message.

31. The electronic communication method of claim **28**, wherein the first message includes information indicating the first party is involved in the message.

32. The electronic communication method of claim **28**, wherein the second message has a portion unpredictable to the second party.

33. The electronic communication method of claim **28**, wherein the second message includes a valid signature of the first party of information representative of the first message.

34. An electronic communication method comprising:

receiving by a first party a first message from a second party without intervention of a trusted third party, at least some of the contents of the first message being unintelligible to the first party;

sending by the first party a second message to the trusted third party, the second message verifying that the first party received the first message; and

receiving by the first party a third message from the trusted third party, the third message enabling the first party to understand the contents of the first message unintelligible to the first party;

wherein the at least some of the contents are able to be rendered intelligible to the first party through assistance of the second party, and

the at least some of the contents are able to be rendered intelligible to the first party through assistance of the trusted third party.

35. The electronic communication method of claim **34**, wherein the first message is based on information not known by the first party.

36. The electronic communication method of claim **34**, wherein at least a part of the second message is unpredictable to the trusted third party.

37. The electronic communication method of claim **34**, wherein the second message is based on information not known by the trusted third party.

38. The electronic communication method of claim **34**, further comprising sending, by the first party after receiving the first message and before receiving the trusted third message, a fourth message to the second party.

39. The electronic communication method of claim **38**, wherein:

the fourth message verifies receipt of the first message by the first party; and

the first party receives no other messages from the second party before receiving the trusted third message.

40. The electronic communication method of claim **38**, wherein the fourth message does not verify receipt of the first message by the first party.

41. The electronic communication method of claim **38**, wherein the fourth message does not verify receipt of the first message by the first party because it does not include a valid signature by the second party of information representative of the first message.

42. The electronic communication method of claim **38**, further comprising receiving, by the first party after sending the fourth message and before receiving the trusted third message, a fifth message from the second party, where the fifth message does not satisfy a predetermined criterion.

43. The electronic communication method of claim **42**, wherein the predetermined criterion is that the fifth message does not include the contents of the first message unintelligible to the second party.

44. The electronic communication method of claim **42**, wherein the predetermined criterion is that the fifth message does not include a valid signature of the contents of the first message unintelligible to the second party.

45. An electronic communication method comprising:

receiving by a trusted first party a first message from a second party verifying that the second party received a second message from a third party without intervention of the trusted first party, the second message including a portion unintelligible to the second party;

sending by the trusted first party a trusted third message to the third party, the trusted third message verifying that the second party received the second message from the third party; and

sending by the trusted first party a trusted fourth message to the second party, the trusted fourth message enabling intelligible disclosure to the second party of the portion of the second message unintelligible to the second party;

wherein the portion is able to be rendered intelligible to the second party through assistance of the third party.

46. The electronic communication method of claim **45**, wherein at least a portion of the first message is unintelligible to the trusted first party.

47. The electronic communication method of claim **45**, wherein the first message is based on information not known to the trusted first party.

48. The electronic communication method of claim **45**, wherein the first message includes a digital signature of the third party.

49. The electronic communication method of claim **48**, wherein the digital signature of the third party signs at least a portion of a message from the second party to the third party.

50. The electronic communication method of claim **45**, wherein at least a portion of the trusted third message is unpredictable to the trusted first party.

51. The electronic communication method of claim **45**, wherein the trusted third message is based on information not known by the trusted first party.

52. The electronic communication method of claim **45**, wherein at least a portion of the trusted fourth message is unintelligible to the trusted first party.

53. The electronic communication method of claim **45**, wherein at least a portion of the trusted fourth message is based on information not known to the trusted first party.

54. A method of processing an electronic message comprising:

processing an original message with a first key to produce a first processed message readable only by a party having a second key;

customizing the first processed message to indicate the party having the second key as a recipient thereof, thereby producing a customized first processed message; and

encrypting the customized first processed message with a third key to produce a second processed message readable only by a party having a fourth key.

55. The method of claim **54**, wherein the first and second keys are different from one another.

56. The method of claim **54**, wherein the third and fourth keys are different from one another.

57. The method of claim **54**, wherein customizing comprises customizing the first processed message to indicate a party producing the first processed message as a sender thereof.

58. The method of claim **54**, wherein the party having the fourth key is a trusted party.

59. The method of claim **54**, further comprising processing the second encrypted message with a fifth key to produce a third processed message.

60. The method of claim **59**, wherein the party having the fifth key is the party having the second key.

61. The method of claim **60**, wherein the second key and the fifth key are identical.

62. The method of claim **59**, where the third processed message uniquely identifies a party which produces it.

63. The method of claim **62**, wherein the party which generates the third processed message is the party having the second key.

64. The method of claim **59**, further comprising processing the third processed message with the sixth key to produce the second processed message.

65. The method of claim **64**, wherein the third processed message is processed by a trusted party.

66. The method of claim **64**, further comprising processing the second processed message, produced by processing the third processed message, with the fourth key to produce the first processed message.

67. The method of claim **66**, wherein the second processed message, produced by processing the third processed message, is processed by a trusted party.

68. The method of claim **59**, wherein message processed using the first key are readable using the fifth key.

69. A method of processing an electronic message comprising:

processing a first processed message, readable only by a party having a first key, with a second key to produce a second processed message uniquely identifying the party processing the first processed message;

wherein the first processed message is a representation of an original message processed with a third key to obtain a third message, readable only to a party having a fourth key, which is further processed with a fifth key to obtain the first processed message.

70. The method of claim **69**, wherein the party processing the first processed message is the party having the fourth key.

71. The method of claim **70**, wherein messages processed using the second key are readable by processing them with the third key.

72. The method of claim **69**, wherein the party having the first key is a trusted party.

73. The method of claim **69**, further comprising processing the second processed message using a sixth key to obtain the first processed message.

74. The method of claim **73**, wherein the second processed message is processed by a trusted party.

75. The method of claim **73**, wherein messages processed using the second key are readable using the sixth key.

* * * * *

US006137884A

United States Patent [19]

Micali

[11] Patent Number: 6,137,884

[45] Date of Patent: *Oct. 24, 2000

[54] **SIMULTANEOUS ELECTRONIC TRANSACTIONS WITH VISIBLE TRUSTED PARTIES**

[75] Inventor: **Silvio Micali**, Brookline, Mass.

[73] Assignee: **Bankers Trust Corporation**, New York, N.Y.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/850,399**

[22] Filed: **May 2, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/700,270, Aug. 20, 1996, Pat. No. 5,629,982, which is a continuation of application No. 08/511,518, Aug. 4, 1995, Pat. No. 5,553,145, which is a continuation-in-part of application No. 08/408,551, Mar. 21, 1995, abandoned.

[51] **Int. Cl.**[7] **H04L 9/00**

[52] **U.S. Cl.** **380/30**; 380/25

[58] **Field of Search** 380/25, 30

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,200,770	4/1980	Hellman et al.	380/30
4,218,582	8/1980	Hellman et al.	380/30
4,405,829	9/1983	Rivest et al.	380/30
4,438,824	3/1984	Mueller-Schloer	380/30
4,458,109	7/1984	Mueller-Schloer	380/30
4,789,928	12/1988	Fujisaki .	
4,885,777	12/1989	Takaragi et al.	380/30
4,885,789	12/1989	Burger et al.	380/25
4,953,209	8/1990	Ryder, Sr. et al.	380/25
5,117,358	5/1992	Winkler .	
5,202,977	4/1993	Pasetes, Jr. et al. .	
5,214,700	5/1993	Pinkas et al.	380/30
5,220,501	6/1993	Lawlor et al. .	
5,243,515	9/1993	Lee .	
5,276,737	1/1994	Micali	380/30
5,315,658	5/1994	Micali	380/30
5,440,634	8/1995	Jones et al.	380/30
5,453,601	9/1995	Rosen .	
5,455,407	10/1995	Rosen .	
5,497,421	3/1996	Kaufman et al.	380/30
5,509,071	4/1996	Petrie, Jr. et al.	380/30
5,553,145	9/1996	Micali	380/30
5,610,982	3/1997	Micali	380/30
5,666,420	9/1997	Micali	380/30

OTHER PUBLICATIONS

Abad–Peiro et al., "Designing a Generic Payment Service" (Nov. 26, 1996).

Asokan et al., "Optimistic Protocols for Multi–Party Fair Exchange," IBM Research Report RZ 2892 (Dec. 9, 1996).

Asokan et al., "Optimistic Fair Exchange of Digital Signatures," IBM Research Report.

Asokan et al., "The State of the Art in Electronic Payment Systems," IEEE Computer, Sep. 1997, pp. 28–35.

Asokan et al., "Optimistic Fair Exchange of Digital signatures" Advances in Cryptology (K. Nyberg, ed.), Proc. Eurocrypt 198 , pp. 591–606 (1997).

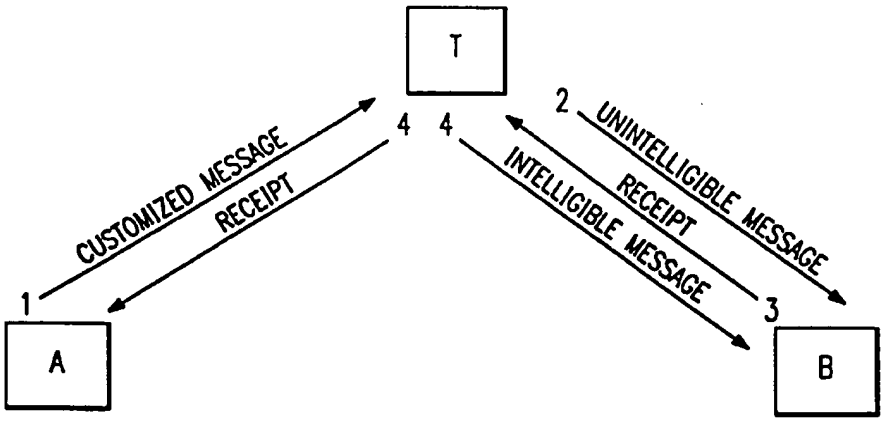
(List continued on next page.)

Primary Examiner—Salvatore Cangialosi
Attorney, Agent, or Firm—Pillsbury Madison & Surto LLP

[57] **ABSTRACT**

A number of electronic communications methods are described involving a first and a second party (i.e., sender and recipient), with assistance from at least a trusted party, enabling electronic transactions in which the first party has a message for the second party. The first party, the second party and the trusted party undertake an exchange of transmissions, such that if all transmissions reach their destinations the second party only receives the message if the first party receives at least one receipt. Preferably, the identity of the first party is temporarily withheld from the second party during the transaction. At least one receipt received by the first party enables the first party to prove the content of the message received by the second party.

29 Claims, 1 Drawing Sheet-



OTHER PUBLICATIONS

Asokan et al., "Optimistic Protocols for Fair Exchange," IBM Research Report RZ 2858 (Sep. 2, 1996).

Asokan et al., "Server–Supported Signatures," Proceedings of ESORICS '96 (Sep. 25–27, 1996).

Asokan et al., "Server–Supported Signatures," Journal of Computer Security pp. 1–13 (1997).

Asokan et al., "Asynchronous Protocols for Optimistic Fair Exchange," IBM Research Report, Proc. IEEE Symposium on Research in Security and Privacy, pp. 86–99 (1998).

Baetlaan et al., "Internet Billing Service Design and Prototype Implementation," Carnegie Mellon University Information Networking Institute 1992 Final Project (Mar. 30, 1993).

Bellare et al., "iKP—A Family of Secure Electronic Payment Protocols" (Jul. 12, 1995).

Ben–Or et al., "A Fair Protocol for Signing Contracts," IEEE Transactions on Information Theory, v. 36 n. 1, pp. 40–46 (Jan. 1990).

Ben–Or et al., "A Fair Protocol for Signing Contracts," Automata, Languages and Programming, pp. 43–52 (Jul. 1985).

Blum, M., "How to Exchange (Secret) Keys," ACM Transactions on Computer Systems, v. 1 n. 2, pp. 175–193 (May 1983).

Burk et al., "Digital Payment Systems Enabling Security and Unobservability," Computers & Security, v. 8, pp. 399–416 (1989).

Burk et al., "Value Exchange Systems Enabling Security and Unobservability," Computers & Security, v. 9, pp. 715–720 (1990).

Camenisch et al., "Digital Payment Systems with Passive Anonymity–Revoking Trustees" Journal of Computer Security (1996).

Camensich et al., "An Efficient Fair Payment System," Proc. 3rd ACM Conf. on Computer Security, pp. 88–94 (1996).

Casey et al., "Secure Automated Document Delivery," Fifth Annual Computer Security Applications Conference, pp. 348–356 (Dec. 4–8, 1989).

Chaum, D., *Security Without Identification: Transaction Systems to Make Big Brother Obsolete,* Comm of ACM, vol. 28 No. 10, pp. 1030–1044 (Oct. 1985).

Chaum, et al., *Untraceable Electronic Cash,* Proc. Crypto '88, pp. 329–327 (1988).

Cheng, et al., *Design and Implementation of Modular Key Management Protocol and IP Secure Tunnel on AIX,* IBM Thomas J. Watson Research Center (Apr. 28, 1995).

Chor, et al., *Verifiable Secret Sharing and Achieving Simultaneity in the Present of Faults,* POC, 26th FOCS, pp. 383–395.

Damgard, I., *Payment Systems and Credential Mechanisms With Provable Security Against Abuse by Individuals,* Proc Crypto '88, pp. 328–335 (1988).

DeMillo, et al., *Protocols for Data Security,* IEEE Computer, pp. 39–50 (Feb. 1983).

Desmedt, et al., *Threshold Cryptosystems,* University of Wisconsin—Milwaukee, pp. 307–315.

Dolev, et al., *Non–Malleable Cryptography,* Comm. of ACM, pp. 542–552 (Mar. 1991).

Dukach, S., *SNPP: A Simple Network Payment Protocol,* M.I.T. Laboratory for Computer Science.

Even, et al., *A Randomized Protocol for Signing Contracts,* Comm. of the ACM, vol. 28, No. 6, pp. 637–647 (Jun. 1995).

Even, S., *Secure Off–Line Electronic Fund Transfer Between Nontrusting Parties,* Computer Science Department Technion, Israel Institute of Technology, pp. 1–10 (Jan. 31, 1988).

Even, et al., *On–Line/Off–Line Digital Signatures,* International Association for Cryptographic Research, 1996, pp. 0–28.

Frankel, et al., *Indirect Discourse Proofs: Achieving Efficient Fair Off–Line E–Cash.*

Franklin, et al., *Fair Exchange with a Semi–Trusted Third Party,* Proc. of the 4th ACM Conf. on Computer and Comm. Security, Apr. 1997, pp. 1–6.

Goldreich, et al., *How to Play Any Mental Game,* Proc. 27th Ann. IEEE ACM Symposium on Theory of Computing, pp. 218–229 (1987).

Goldreich, et al., *Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero–Knowledge Proof Systems,* Association for Computing Machinery, vol. 38, No. 1, pp. 691–729 (Jul. 1991).

Goldwasser, et al., *The Knowledge Complexity of Interactive Proof Systems,* SIAM J. Comput. vol. 18, No. 1, pp. 186–208 (Feb. 1989).

Herda, S., *Consulting evidence and proof in digital cooperation,* Computer Standards and Interfaces 17 (1995), pp. 69–79.

Hickman, et al., The *SSL Protocol,* Netscape Communications Corp. (Jun. 1995).

Jakobsson, M., *Reducing Costs in Identification protocols,* Crypto '92 (1992).

Janson, et al., *Electronic Payment Over Open Networks,* IBM Zurich Research Laboratory CH 8803 Ruschlikon, Switzerland (Apr. 18, 1995).

Janson, et al., *Electronic Payment Systems,* pp. 1–24 (May 1, 1996).

Kilian, J., et al., *Identity Escrow,* pp. 1–18.

Koleta, G.B., Cryptographers Gather to Discuss Research, Science, pp. 646–647 (Nov. 11, 1981).

Konheim et al., *Digital Signatures and Authentications,* Cryptography, A Primer, (1981), pp. 334–367.

Low, et al., *Anonymous Credit Cards,* 2nd ACM Conference on Computer and Communication Security, pp. 1–10 (1994).

Luby, et al., *How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically–Biased Coin,* IEEE, pp. 11–21 (1983).

Myer, P., *Cryptography: A guide for the design and implementation of crytptographic Systems,* McGraw–Hill, Inc., pp. 386–430 (1982).

Mueller–Schloer, et al., *The Implementation of a Cryptography–Based Secure Office System,* AFIPS Conference Proc. 1982, pp. 487–492 (1982).

Needham, et al., *Using Encryption for Authentication in Large Networks of Computers,* Comm. of ACM, vol. 21, No. 12, pp. 993–999 (Dec. 1978).

Pedersen, T., *Electronic Payments of Small Amounts,* Aarhus Univ. Tech. Rpt. DAIMI PB–495, pp. 1–12 (Aug. 1995).

Otway, et al., *Efficient and Timely Mutual Authentication,* ACM Operating Systems Review, vol. 21, No. 1, pp. 8–10 (Jan. 1987).

Neuman, et al., *Requirements for Network Payment: The NetCheque™ Perspective,* Proc. IEEE Compcon '95, San Francisco (Mar. 1995).

Rabin, M., *How To Exchange Secrets,* (May 20, 1981) pp. 1–21.

Rabin, M., *Transaction Protection by Beacons,* TR–29–81, Harvard University Center for Research in Computing Technology, (Nov. 1981) pp. 1–21.

Rescorla et al., *The Secure HyperText Transfer Protocol,* Enterprise Integration Technologies, (Jul. 1995) pp. 1–40.

Rihaczek, K., *Teletrust,* Computer Networks and ISDN Systems 13 (1987) pp. 235–239.

Rivest et al., *A Method for Obtaining Digital Signatures and Public–Key Cryptosystems,* Programming Techniques (Feb. 1978) pp. 120–126.

SEMPER Project AC026 ACTS Programme (Mar./Jul. 1995).

Serenelli et al., *Securing Electronic Mail Systems, MILCOM 92* (San Diego, CA 1992) pp. 29.1.1–29.1.4.

Shamir, A., *How to Share a Secret,* Comm. ACM v. 22, n. 11 (Nov. 1979) pp. 612–613.

Simmons, J., *An Impersonation–Proof Identity Verification Scheme,* Advances in Cryptology—CRYPTO '87, pp. 211–215.

Simmons, J., *Zero–Knowledge Proofs of Identity and Veracity of Transaction Receipts,* Advances in Cryptology—EUROCRYPT 188, pp. 35–49.

Simmons, *A Protocol to Provide Verifiable Proof of Identity and Unforgeable Transaction Receipts,* IEEE Journal on Selected Areas in Communications, vol. 7, No. 4, May 1989, pp. 435–447.

Sirbu, et al., *NetBill: An Internet Commerce System Optimized for Network Delivered Services,* Engineering and Public Policy Department, Computer Science Department, Carnegie Mellon. University, Pittsburgh, Pennsylvania 15213, pp. 1–11.

Snare, J.L., *Secure Electronic Data Interchange,* Computer Security in the Age of Information, (W.L. Caelli, ed.), IFIP, 1989, pp. 331–342, (1989).

Sollins, K.R., *Cascaded Authentication,* IEEE Symposium on Security and Privacy (Apr. 18–21, 1988), pp. 156–163.

Stadler et al, *Fair Blind Signatures,* Advances in Cryptology—EUROCRYPT '95 (1995).

Stein et al., *The Green Commerce Model,* pp. 1–17, (Oct. 1994).

Tsudik, G., *Zurich iKP Prototype (ZiP), Protocol Specification Document,* IBM Zurich Research, pp. i–27 (Mar. 5, 1996).

Varadharajan et al., *Formal Specification of A Secure Distributed Messaging System,* 12th National Computer Security Conference Proceedings, pp. 146–171, (Oct. 1991).

Varadharajan, V., Notification: *A Practical Security Problem in Distributed Systems,* 14th National Computer Security Conference, pp. 386–396, (Oct. 1991).

Waidner, M., *Development of a Secure Electronic Marketplace for Europe,* Proc. of ESORICS 96, Rome, (Seq. 1996), pp. 1–15.

Zhou et al., A Fair Non–Repudiation Protocol, IEEE (1996), pp. 55–61.

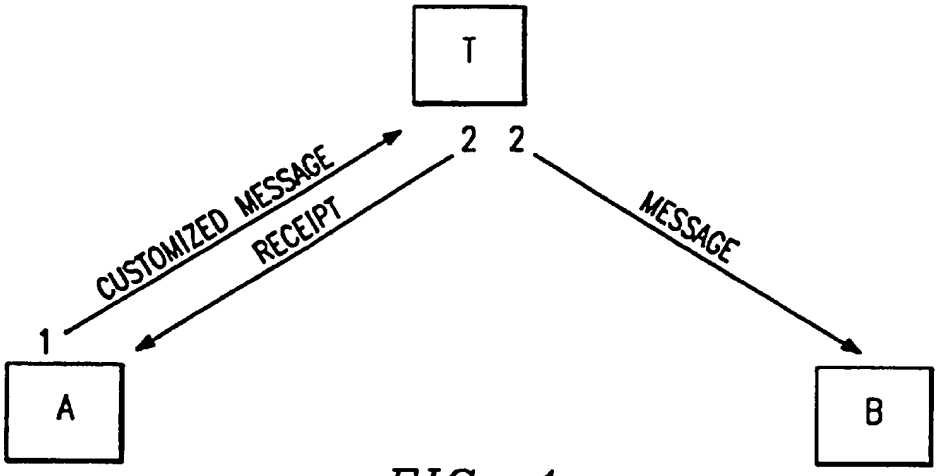

T
2 2
CUSTOMIZED MESSAGE
RECEIPT
MESSAGE
1
A
B

FIG. 1

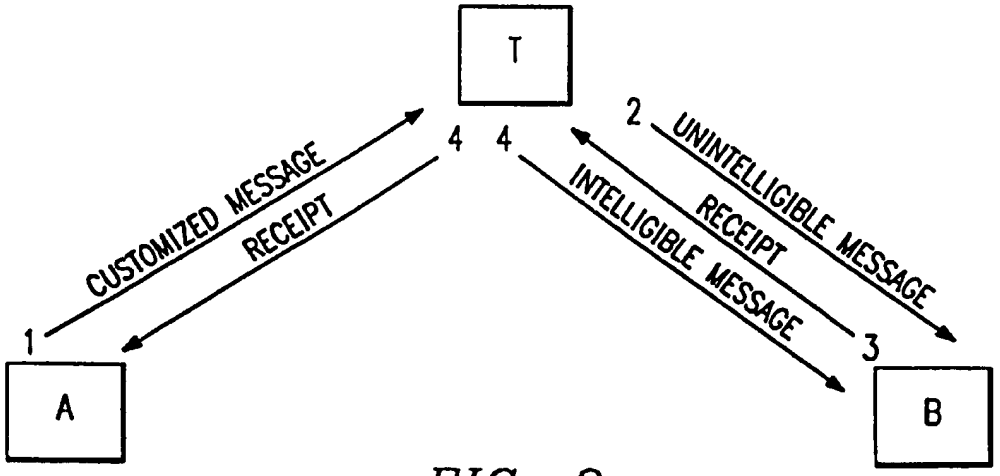

T
4 4
2
UNINTELLIGIBLE MESSAGE
CUSTOMIZED MESSAGE
RECEIPT
RECEIPT
INTELLIGIBLE MESSAGE
3
1
A
B

FIG. 2

SIMULTANEOUS ELECTRONIC TRANSACTIONS WITH VISIBLE TRUSTED PARTIES

RELATED APPLICATION

This application is a continuation of Ser. No. 08/700,270, filed Aug. 20, 1996, now U.S. Pat. No. 5,629,982, which is a a continuation of application Ser. No. 08/511,518 filed on Aug. 4, 1995 now U.S. Pat. No. 5,553,145, which is a continuation-in-part of prior application Ser. No. 08/408,551, filed Mar. 21, 1995 now abandoned.

TECHNICAL FIELD

The present invention relates generally to electronic commerce and transactions and more particularly to techniques for enabling users to effect certified mail, contract signing and other electronic notarization functions.

BACKGROUND OF THE INVENTION

The value of many transactions depends crucially on their simultaneity. Indeed, simultaneity may be so important to certain financial transactions that entities often are willing to incur great inconvenience and expense to achieve it. For example, consider the situation where two parties have negotiated an important contract that they now intend to "close." Often, the parties find it necessary to sign the document simultaneously, and thus they meet in the same place to watch each other's actions. Another example is the process of certified mail, where ideally the sender of a message desires that the recipient get the message simultaneously with the sender's obtaining a "receipt". A common certified mail procedure requires a person who delivers the mail to personally reach the recipient and obtain a signed acknowledgment when the message is delivered. This acknowledgment is then shipped to the sender. Again, this practice is costly and time consuming. Moreover, such acknowledgments do not indicate the content of the message.

In recent years, the cost, efficiency and convenience of many transactions have been improved tremendously by the availability of electronic networks, such as computer, telephone, fax, broadcasting and others. Yet more recently, digital signatures and public-key encryption have added much needed security to these electronic networks, making such communication channels particularly suitable for financial transactions. Nevertheless, while electronic communications provide speed, they do not address simultaneity.

The absence of simultaneity from electronic transactions severally limits electronic commerce. In particular, heretofore there has been no effective way of building so-called simultaneous electronic transactions ("SET's"). As used herein, a SET is an electronic transaction that is simultaneous at least in a "logically equivalent" way, namely it is guaranteed that certain actions will take place if and only if certain other actions take place. One desirable SET would be certified mail, however, the prior art has not addressed this problem effectively. This can be seen by the following consideration of a hypothetical example, called ideal certified mail or "ICM". In an ICM transaction, there is a sender, Alice, who wishes to deliver a given message to an intended recipient, Bob. This delivery should satisfy three main properties. First, Bob cannot refuse to receive the message. Second Alice gets a receipt for the message if and only if Bob gets the message. Third, Alice's receipt should not be "generic," but closely related to the message itself. Simultaneity is important in this transaction. For instance, Alice's message could be an electronic payment to Bob, and it is desired that she obtains a simultaneous receipt if possible.

Alice could try to get a receipt from Bob of a message m in the following way. Clearly, sending m to Bob in the clear as her first communication does not work. Should this message be her digital signature of an electronic payment, a malicious Bob may lose any interest in continuing the conversation so as to deprive Alice of her receipt. On the other hand, asking Bob to send first a "blind" receipt may not be acceptable to him.

Another alternative is that Alice first sends Bob an encryption of m. Second, Bob sends Alice his digital signature of this ciphertext as an "intermediate" receipt. Third, Alice sends him the decryption key. Fourth, Bob sends Alice a receipt for this key. Unfortunately, even this transaction is not secure, because Bob, after learning the message when receiving Alice's key, may refuse to send her any receipt. (On the other hand, one cannot consider Bob's signature of the encrypted message as a valid receipt, because Alice may never send him the decryption key.)

These problems do not disappear by simply adding a few more rounds of communication, typically consisting of "acknowledgments". Usually, such additional rounds make it more difficult to see where the lack of simultaneity lies, but they do not solve the problems.

Various cryptographic approaches exist in the literature that attempt to solve similar problems, but they are not satisfactory in many respects. Some of these methods applicable to multi-party scenarios propose use of verifiable secret sharing (see, for example, Chor et al), or multi-party protocols (as envisioned by Goldreich et al) for making simultaneous some specific transactions between parties. Unfortunately, these methods require a plurality of parties, the majority of which are honest. Thus, they do not envision simultaneous transactions involving only two parties. Indeed, if the majority of two parties are honest then both parties are honest, and thus simultaneity would not be a problem. Moreover, even in a multi-party situation, the complexity of these prior art methods and their amount and type of communication (typically, they use several rounds of broadcasting), make them generally impractical.

Sophisticated cryptographic transactions between just two parties have been developed but these also are not simultaneous. Indeed, if just two people send each other strings back and forth, and each one of them expects to compute his own result from this conversation, the first to obtain the desired result may stop all communications, thereby depriving the other of his or her result. Nonetheless, attempts at providing simultaneity for two-party transactions have been made, but by using assumptions or methods that are unsatisfactory in various ways.

For example, Blum describes transactions that include contract signing and certified mail and that relies on the two parties having roughly equal computing power or knowledge of algorithms. These assumptions, however, do not always hold and are hard to check or enforce anyway. In addition, others have discovered ways to attack this rather complex method. A similar approach to simultaneity has also been proposed by Even Goldreich and Lempel. In another Blum method for achieving simultaneous certified mail, Alice does not know whether she got a valid receipt. She must go to court to determine this, and this is undesirable as well.

A method of Luby et al allows two parties to exchange the decryption of two given ciphertexts in a special way, namely,

for both parties the probability that one has to guess correctly the cleartext of the other is slowly increased towards 100%. This method, however, does not enable the parties to achieve guaranteed simultaneity if one party learns the cleartext of the other's ciphertext with absolute certainty (e.g., by obtaining the decryption key); then he can deny the other a similar success.

For this reasons several researchers have tried to make simultaneous two-party transactions via the help of one or more external entities, often referred to as "centers", "servers" or "trustees", a notion that appears in a variety of cryptographic contexts (see, for instance, Needham and Schroder and Shamir). A method for simultaneous contract signing and other transactions involving one trustee (called a "judge") has been proposed by Ben-Or et al. Their method relies on an external entity only if one party acts dishonestly, but it does not provide guaranteed simultaneity. In that technique, an honest party is not guaranteed to have a signed contract, even with the help of the external entity. Ben-Or et al only guarantee that the probability that one party gets a signed contract while the other does not is small. The smaller this probability, the more the parties must exchange messages back and forth. In still another method, Rabin envisions transactions with the help of external party that is active at all times (even when no transaction is going on), but also this method does not provide guaranteed simultaneity.

The prior art also suggests abstractly that if one could construct a true simultaneous transaction (e.g., extended certified mail), then the solution thereto might also be useful for constructing other types of electronic transactions (e.g., contract signing). As noted above, however, the art lacks an adequate teaching of how to construct an adequate simultaneous transaction.

There has thus been a long-felt need in the art to overcome these and other problems associated with electronic transactions.

BRIEF SUMMARY OF THE INVENTION

It is an object of the invention to provide true simultaneous electronic transactions.

It is a further object of the invention to provide electronic transactions having guaranteed simultaneity in a two-party scenario with the assistance of a visible trusted party.

It is another more specific object of the invention to provide ideal certified mail wherein the identity of the sender is temporarily withheld from the recipient during the transaction.

It is still another object of the invention to provide a simultaneous electronic transaction wherein the recipient can prove the content of a message and a receipt provided to the sender proves the content of the message.

These and other objects are provided in an electronic communications method between a first and a second party, with assistance from at least a trusted party, enabling an electronic transaction in which the first party has a message for the second party. A first method, called the sending receipt approach, begins by having the first party transmit to the trusted party a custom version of the message intelligible to the second party but not by the trusted party. In response, the method continues having the trusted party verify that the first party transmitted the custom version of the message and that the second party is the intended recipient thereof. The trusted party then transmits to the second party information from which the second party can retrieve the message. Then, the trusted party transmits to the first party a sending receipt indicating that the message has been transmitted to the second party. At least one of the transmissions is carried out electronically.

According to an alternative embodiment, called the return receipt approach, the method begins having the first party transmit to the trusted party a custom version of the message intelligible to the second party but not by the trusted party. In response, the method continues by having the trusted party verify that the first party transmitted the custom version of the message and that the second party is the intended recipient thereof. The trusted party then transmit to the second party first information which determines the message but retains the message and the identity of the first party hidden from the second party. A test is then done to determine whether within a given time the second party transmits to the trusted party a return receipt indicating that the second party received the transmission of the first information from the trusted party. If the second party transmits the return receipt to the trusted party, the method has the trusted party (i) transmit to the second party second information from which the second party, using the first and second information, can retrieve the message, and (ii) transmit to the first party a receipt that the second party has received the message. Again, at least one of the transmissions is carried out electronically.

Many other electronic communications methods are described wherein the first party, the second party and the trusted party undertake an exchange of transmissions, at least one of which occurs electronically and in an encrypted manner, such that if all transmissions reach their destinations the second party only receives the message if the first party receives at least one receipt. At least one receipt received by the first party enables the first party to prove the content of the message received by the second party.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference should be made to the following Detailed Description in conjunction with the accompanying drawings in which:

FIG. 1 illustrates a preferred sending receipt method of the invention; and

FIG. 2 illustrates a preferred return receipt method of the invention.

DETAILED DESCRIPTION

In each of the schemes described below, there is a user Alice and a user Bob. The trusted party may be a financial center that facilitates SETs among its customers, including Alice and Bob. For convenience, the following description shows how to make extended certified mail "simultaneous", although the invention is not so limited. In the context of an ICM system, the third party is called the Post Office. The inventive scheme is also preferable to ordinary certified mail because the message receipt also guarantees the content of the message. Also, the electronic transaction is faster, more informative and more convenient than traditional certified mail, and its cost should be substantially lower.

In the preferred embodiment, an extended certified mail system is provided using a single "trusted" party. The system is implemented in a computer network, although it should be realized that telephone, fax, broadcast or other communication networks may be used. Thus, without limitation, it is assumed that each user in the system has a computer capable of sending and receiving messages to and from other computers via proper communication channels.

Each user in the system has a unique identifier. Alice's identifier is denoted by A, and Bob's identifier is B. The identifier of the Post Office is denoted by PO. Users and the Post Office can digitally sign messages. Thus, each has a secret signing key and a matching public verification key. If m is a message (string), then $SIG_A(m)$ indicates Alice's signature of m. (It is assumed, for convenience, that m is always retrievable from its signature. This is the case for most signature schemes, and it is otherwise possible to consider a signed message as the pair consisting of the message and its signature.)

Users and the Post Office can encrypt messages by means of a public-key encryption algorithm (e.g., RSA). Thus, each has a public encryption key and a corresponding secret decryption key. $E_A(m)$, $E_B(m)$, and $E_{PO}(m)$ denote, respectively, the encryption of a message m with the public key of Alice, Bob, and the Post Office. For simplicity, it is assumed that these schemes are secure in the sense that each of EA', EB' and E_{PO} appear to behave as a random function. The system can be suitably modified if these functions are much less secure.

Again, for simplicity these encryption algorithms are deterministic and uniquely decodable. Thus, given a value y and a message m, all can verify whether y is the encryption of m with, for example, the Post Office's key, by checking whether $E_{PO}(m)$ equals y. (If the encryption scheme is probabilistic, then one may convince another that a string y is an encryption of a message m by providing m together with the random bits that were used to encrypt m.) (It may also be possible to use encryption algorithms that are not uniquely decodable, for instance, if it is hard to decrypt a given ciphertext in two different ways.) For simplicity, if public key encryption algorithms are used, messages are encrypted directly with a public-key algorithm, however, one could first encrypt a message conventionally with some key k, and then encrypt k with a public-key algorithm. (Thus, to decrypt m, one need only just decrypt k). Indeed, private key encryption algorithms could be used throughout.

According to the invention, it is desired to devise practical [CM methods, involving more visible trustees, that (1) produce receipts closely tied to the content of the mail, (2) hide (at least temporarily) the identity of senders from the recipients, and (3) can be implemented in a pure electronic manner (at least, as long as senders and recipients behave properly).

THE SENDING-RECEIPT METHOD

To describe the various methods of the present invention, assume there are senders, receivers and post offices. It should be clear, however, that each of these may be any entity, such as a person, a person's representative, a physical device (in particular, a tamper-proof device) or a collection of people and/or physical devices. For example, the Post Office could be a tamper-proof device located in a device or facility belonging to Alice and/or Bob.

Also, in the preferred embodiments, Alice, Bob and the Post Office all have public encryption keys and matching secret decryption keys (e.g. like in the RSA algorithm), that their cryptosystem behave like random functions, and that they can digitally sign messages (preferably by an algorithm different than their encryption one). An encryption of a string s with the public key of Alice, Bob, and the Post Office will be denoted, respectively, as $E_A(s)$, $E_B(s)$, $E_{PO}(s)$. The digital signature of a string s by Alice, Bob, and the Post Office will, respectively, be denoted by $SIG_A(s)$ $SIG_B(s)$, and $SIG_{PO}(s)$. (it is understood that messages can be one-way hashed prior to being signed, together with other valuable information, such as recipient, time, transaction type, sender and recipient, etc.) Identifiers for Alice, Bob, and the Post Office will, respectively, be denoted by A, B, and PO.

In the present invention, a customization step is used by Alice to identify (usually to the Post Office) herself as the sender and Bob the (ultimate) recipient of some string s (usually a message m encrypted with Bob's public encryption key). This step prevents cheating. In particular, it prevents an enemy from sending to Bob the same message Alice does and in a certified manner. Any customization step is in the scope of the present invention. A simple such step consists of having Alice send the Post Office a value $z=E_{PO}(A, B, E_B(m))$. Indeed, should the Post Office receive from some user X other than Alice the value z, upon decrypting it with its secret decryption key, it will compute $(A, B, E_B(m))$ and thus realize that there is a problem with the identity of the sender.

The above customization works well if the encryption function behaves as a random function. Alternative and more sophisticated customizations, all within the scope of the invention, are also possible. For instance, Alice may send the Post Office $z=E_{PO}(SIG_A(ICM, B, E_B(m)))$, where the identifier ICM signifies that z is part of an electronic certified mail transaction. Such identifiers may be dismissed, particularly if standard formats are adopted for ICM transactions. As another example, Alice may achieve customization by using identifiers and her digital signature both outside and inside the Post Office's encryption layer: $z=SIG_A(A, B, E_{PO}(SIG_A(A, B, E_B(m))))$. In some contexts (e.g., but without limitation, when the communications channel is believed to be secure), it may suffice to use a customization where the identity of the sender and the message are sent separately, whether or not signed together (e.g., $(B, E_B(m))$ or $SIG_A(B, E_B(m))$).

The basic electronic certified mail system with a visible party is now described. At least one transmission in the method below (and preferably all) are electronic, where by "electronic" we mean any non-physical delivery, including, without limitation, transmissions via telephones, computer networks, radio, broadcasting, air waves, and the like.

THE BASIC METHOD

A1 Sender Step): Let m be the message that Alice desires to send Bob by certified mail. Then Alice sends to the Post Office a customized version of m that is intelligible by Bob, but not by the Post Office.
(e.g., she sends the value $z=E_{PO}(A, B, E_B(m))$.
Preferably, Alice's communication is digitally signed and indicates, in a standard manner, that it should be delivered certified to Bob. (e.g., using an alternative customization step, just for illustration purposes, she sends $z=E_{PO}(SIG_A(ICM, B, E_B(m)))$, or $E_{PO}(SIG_A(B, E_B(m)))$.) It is also preferable that Alice specifies additional valuable information, such as time information and information easily alerting the Post Office that her transmission is part of an ICM transaction.

PO1 (Post Office Step): After receiving Alice's transmission, the Post Office preferably uses the customization step to verify that Alice is the sender and Bob the intended recipient of this piece of electronic certified mail. If this is the case, then it sends to Bob information enabling him to retrieve Alice's message, preferably using digital signatures, and indicating to him but hiding from others that it is a piece of ICM

from Alice to him, (e.g., it sends $y=E_B(SIG_{PO}(ICM, A, B, E_B(m)))$, or ICM, y, so that Bob it is more easily alerted that he is dealing with an ICM transaction).

If Alice has made use of digital signatures (e.g., if she has signed $E_B(m)$ or a value comprising it in Step AI, then it is preferable that these signatures are also forwarded to Bob. (e.g., if Alice sent the Post Office the value $SIG_A(EB(m))$ as part of her Step AI, then the Post Office may send $E_B(SIG_{PO}(ICM, A, B, SIG_A(E_B(m))))$ to Bob in this step.)

In addition, the Post Office also sends Alice her receipt. Preferably this involves a digital signature that it has sent Alice a message to Bob in a way intelligible to him. Such a receipt preferably also indicates other valuable information, such as the time, T, when this was done. (e.g., it sends Alice $E_A(SIG_{PO}(ICM, A, B, T\ E_B(m)))$.)

The Post Office of the Sending-Receipt Method is visible because it takes part to the transaction whether or not Alice and Bob behave honestly. It should be understood that each party to the transaction (whether the Sending Receipt method or the Return Receipt method or other methods of the invention) may participate in the transaction via a representative. In such case, for instance, Alice may be identified with a representative. Alternatively, it should be understood that a party may only be partially-identified with his own representative. For instance, the message may be sent to Bob's representative but be intelligible only to Bob himself.

The Post Office is not trusted with the knowledge of Alice's (cleartext) message to Bob; indeed, it cannot understand m. It is trusted, instead, to perform a proper delivery, which makes the Sending-Receipt Method a (logically) simultaneous transaction; indeed, Alice gets Bob's receipt if and only if Bob gets information from which he can retrieve Alice's message. The simultaneity of the transaction is not affected by the order in which the Post Office sends the encrypted message to Bob and the receipt to Alice. What matters is that it sends both of them or none, or that functionally equivalent steps are taken to preserve simultaneity.

Alice's receipt certifies that her message was properly sent to Bob, but not the fact that Bob actually received it. The Post Office is indeed trusted with properly sending messages and this can be construed to include that these messages sent by the Post Office reach their destinations. But receiving a piece of mail (i.e. having a letter deposited in the right mailbox or having an electronic message reach the right computer) may not mean that the recipient is aware of the delivery. It is this awareness that is necessary in many scenarios, such as many legal applications. This is why the present method is called a sending-receipt method. The method thus is the electronic equivalent of traditional certified mail, without return receipt.

The electronic nature of the method, however, requires some special care, such as a proper customization step. Indeed, in traditional electronic mail, it is easy to achieve that an enemy cannot send to Bob the same message Alice does, because, if he does not know this message a priori, he is prevented from copying by the envelope containing it. $E_B(m)$, however, is a kind of envelope that prevents understanding m, but can be copied. Indeed, if Alice sends $E_B(m)$ to Bob without customization and an enemy intercepts her transmission, he may easily send the same ciphertext $E_B(m)$ to Bob (by certified mail or not), creating various potential problems. This has been a recognized problem in cryptography in different contexts. Notice that having Alice just sign $E_B(m)$ does not solve the problem. Indeed, an enemy X who captures $SIG_A E_B(m))$, easily learns the value $E_B(m)$ (because signatures generally guarantee the message, but do not hide it), and can then easily sign it himself, that, send $(SIG_X E_B(m))$ as part of his own ICM transaction.

In the present invention, encryption of the message m with a key associated to a party X, $E_X(m)$, should be broadly construed to include any information that enables X (and only X) to retrieve the message m. For instance, $E_X(m)$ may consist of the encryption with a key associated with X of another key with which the message m has already been encrypted. (This other encryption of m may already be in possession of X, or sent separately to X, or publicly-known, or otherwise knowable by X).

The electronic sending-receipt method is more than equivalent to traditional certified mail (without return receipt). Indeed, if digital signatures are properly used as exemplified above, not only does Bob learn (and can prove) Alice's identity and get Alice's message, he can also prove to third parties what this message is. For instance, if the Post Office (in Step PO1), sends him the value $v=(SIG_{PO}(E_B(A, B, E_B(m)))$, if Bob hands out v and m to a third party, the latter can compute $u=E_B(m)$ by means of Bob's public encryption key, and then (again due to Bob's public encryption key) the value $s=E_B(A, B, u)$, and, finally he can verify whether v is the Post Office's digital signature of s. If the Post Office is trusted with respect to deliver just what it is supposed to, then this is sufficient proof that Bob got m from Alice via ICM. Indeed, Alice's message can be defined to be whatever string x can, when encrypted with Bob's key, yields the value $E_B(m)$. If such x is nonsensical, then Alice sent Bob a nonsensical message. This convention prevents Bob from claiming that he did not really get Alice's message in this way.

Should one prefer to trust the Post Office even less, and still enable Bob to prove which message he got from Alice, it suffices, for instance, that Alice makes use of digital signatures; e.g., she sends $z=E_{PO}(SIG_A(ICM, B, E_B(m)))$ in Step A1, and the Post Office sends $SIG_A(ICM, B, E_B(m))$ preferably further signed and encrypted—to Bob in Step PO1. This way, by revealing m, Bob can prove via Alice's signature that she indeed sent him m by extended certified mail.

The electronic sending-receipt method is superior to traditional certified mail in another respect. Alice's receipt needs not to be a generic one, but enables her to prove the exact content of the message she sent Bob. In fact, if her receipt consists of the Post Office's digital signature that it has sent $z=E_{PO}(A, B, E_B(m))$ to Bob, by revealing m she enables anyone to compute $v=E_B(m)$ from Bob's public encryption key, and thus $E_{PO}(A, B, v)$ from the Post Office's public encryption key, so as to verify that the result is indeed z, the value signed by the Post Office.

The ICM is superior to other electronic methods for certified mail in many respects. In particular, simultaneity is guaranteed, rather than being just highly probable. Moreover, since the Post Office provides Alice with her receipt, Bob cannot decide whether or not to accept a message from her based on the sender's identity.

It is recommended that each transmission occur within the encryption layer of its immediate recipient. (e.g., in Step A1, it is preferable that Alice sends $E_{PO}(SIG_A(ICM, B, E_B(m)))$ rather than $SIG_A(ICM, B, E_B(m))$.) Among other things, this way of transmitting denies an enemy monitoring such transmissions valuable information, such as sender-receiver information. That is, if an enemy learns $E_B(SIG_{PO}(ICM, B, E_B(m)))$, the transmission of the Post Office to Bob of Step

PO1, and it further knows that this value was travelling from the Post Office to Bob, it may deduce that Bob is the recipient of a piece of certified mail, but it may not easily learn that the sender was Alice because this piece of data is protected under Bob's encryption key. Indeed, the Post Office may make this harder by processing its PO1 steps relative to different senders and recipients in a different order. If at every time interval there are sufficiently many senders, this will confuse the enemy even more. In addition, the Post Office may arrange for dummy transmissions, so as to have sender traffic that always looks reasonably busy. This enables it to process real and fake sending request in an interwoven order without creating any delays. If desired, however, most recipient-encryption protections could be dispensed with.

Finally, the reference to m as the message Alice wants to send to Bob should be broadly construed to mean any message that Alice has for Bob, including a message that is chosen before the transaction, but arises or is implicitly defined by the transaction.

VARIANTS AND IMPROVEMENTS. Many variants of the above and following methods are applicable and within the scope of the invention. In particular, customization may be dismissed all together or achieved by means of other electronically transmissible methods. The sender's identity may be used for customization purposes, but hidden from the recipient in some applications. Alice's message may not be hidden from the Post Office. (e.g., if this is a machine, or consists of a collection of individuals, many of which must cooperate to learn the message). Also, digital signatures should be broadly construed to include any form of electronically transmissible guarantees. Conventional encryptions may be used in alternative or in conjunction with public-key one. A higher level of interaction may be adopted in our methods (e.g., if one wishes to get additional valuable benefits, such as zero-knowledge). In particular, each of our Steps can be realized by means of more rounds of communications. Time information may be included in some or all of the transmissions, each party may be a multiplicity of parties, and so on.

Proper use of time information may be important. For instance, assume Alice specifies (preferably in an untamperable way) to the Post Office the time in which her string was sent. If the Post Office receives it too late (or too early), it may not send any communication to Bob nor any receipt to Alice. (Indeed, if the certified message from Alice to Bob is an order to buy stock that day, Bob may not be responsible for failing to obey the order if he got it unreasonably late.) Alternatively, the Post Office may specify in its communication to Bob the time when this was sent, preferably in a digitally signed manner, so that, among other things, Bob may in many contexts prove that he got Alice's message too late. The Post Office may also deny Alice her receipt if her AI transmission arrives too late, or it may issue her a properly "time-stamped" receipt, but such receipt may be deemed void for certain purposes if some of the time information indicated is deemed to be too late.

Multiplicities of parties may also be quite useful. For instance, Alice may deal with two or more Post Offices for delivering the same message to Bob. In this case, having two independent receipts for the same message constitutes a much greater evidence that at least one of the Post Offices has properly sent the message to Bob.

Alternatively, Alice may conveniently deal with a single Post Office, but this is an entity comprising or coordinating several agents. Such an entity may give Alice's communication to two or more of its agents, and these will send Alice's message to Bob in the proper manner, generating the proper receipts. These receipts may then be given by the agents to Alice directly, or to the (or some other) entity, who then will give them (or sufficiently many of them, or a consolidated version of some of them) to Alice.

It is also useful that the Post Office agents possess pieces of a secret key of the Post Office. In this case one may wish that they collaborate for decrypting some communications sent to the Post Office in an encrypted manner. If some of these communications are intended for someone else (e.g., if one such communication consists of or includes $E_B(m)$ encrypted with the Post Office' key), then the Post Offices agents may enable directly the recipient to decrypt the communication (e.g., they may enable only Bob to reconstruct $E_B(m)$. This may be achieved, for instance, by a proper use of threshold cryptosystems. Indeed, if single agents are incapable of understanding messages encrypted with the Post Office's key, it may be unnecessary for Alice to first encrypt her message m to Bob with Bob's key. She may directly encrypt m with such a multi-party controlled key of the Post Office, the agents of the Post Office will then enable Bob to decrypt m, while the agents and/or the Post Office will give Alice a proper receipt. A single or sufficiently few agents of the Post Office will not, however, be able to understand m.

Another improvement is the following. In the Sending-Receipt Method Bob may claim that he did not "really" receive Alice's message because he lost his decryption key. To solve this problem, the Post Office may perform the Return Mail Service only for those users who guarantee to back up their secret decryption keys in a deemed acceptable way; so that, for instance, such a Bob may not use his having lost his secret key as a defense against an unwanted piece of certified mail. For example, to be eligible to receive a piece of ICM, it can be required that Bob performs (or that he has have already performed) a given key-escrow procedure relative to his keys used for electronic certified mail purposes. This way, Bob may always be capable of retrieving his secret key.

To create further incentive for Bob to undergo this key-escrow step, it may be stipulated that a user cannot be a sender of an ICM system, unless he also is a potential receiver with a properly backed up key. In any case, the Post Office (or a court if and when it is invoked) may regard Bob as a legitimate receiver if he had given a suitable and timely indication that he accepts a given key of his to be used for ICM purposes.

Alternatively, Bob may be regarded to be a legitimate recipient of a piece of ICM by the mere fact that a key of his is known to be suitably backed up (e.g., by an approved key-escrow method), and it was this key of his to be used as the recipient-key in a ICM transaction. The fact that Bob has elected a key of his to be usable as a recipient-key for ICM purposes, of the fact that a key of his is suitably backed up, may, for instance, be part of a certificate of this key (e.g., of the certificate showing that this key belongs to Bob). Alternatively, Bob may coincide for ICM purposes with a plurality of entities each having a piece of "his" decryption key, so that sufficiently many of these entities may recovery any message encrypted with Bob's encryption key. Thus, the Post Office may communicate with each or sufficiently-many of these entities.

Alternatively, if, as described above, the Post Office has several agents so as to offer a service based on a type of threshold cryptosystem and messages are not further encrypted with a recipient key, there is no worry that the recipient may lose his key. Indeed, it will be the Post Office

who will enable him to get his message from Alice. Notice also that a weaker customization of Alice's message to Bob may be realized within Bob's encryption layer, or even solely within this layer.

For instance, Alice may send to the Post Office $z=E_{PO}(w)$, where $w-E_B(A, B, m)$ or $(w=E_B(SIG_A(m)))$, just to give an example of an alternative customization in this setting. In this setting, the message received by Bob is conventionally declared to be m only if w is an encryption of (A, B, m), that is, if it identifies in some standard way Alice as the sender and Bob as the recipient. For instance, if Bob is a stock-broker and m a purchaser order of a given stock, if v does not consist of A, B, m, Bob is not obliged to buy that stock. This way of proceeding facilitates the job of the Post Office (for instance because it may not be asked to check any customization) and still offers valuable protection.

The Return-Receipt Method

Despite its utility, the Sending-Receipt Method suffers from the following problem: Bob may never receive (or claim not to have received) Alice's (cleartext) message, not because he lost (or claims to have lost) his decryption key, but because he never got (or claims to have not gotten) any communication from the Post Office. For instance, if a computer network is used for communicating during an ICM transaction, a failure may occur or may claimed to have occurred.

To solve such problems, the Sending-Receipt Method is augmented as follows. After receiving the communication of Step PO1, Bob may be asked or required to send a proper receipt back. This receipt may be sent to the Post Office (or directly to Alice, since at that point Bob may have already learned Alice's identity). Such receipt, if obtained, simplifies matters a great deal, and offers much greater guarantees to everyone involved. Upon receiving it, the Post Office may store it, or send it to Alice as an additional receipt, or issue to Alice an equivalent additional receipt.

Alternatively, the Post Office may withhold Alice's receipt of Step PO1, and give it to her only if Bob does not produce any receipt for the Post Office's PO1 transmission to him. Moreover, if Bob does not produce a receipt, the Post Office may take some of the actions described below that enable it to obtain a receipt from Bob in some other manner or enable it to produce a suitable affidavit (e.g., that Bob willingly refused Alice's message). It is expected that Bob will readily acknowledge the Post Office PO1 transmission most of the times. Indeed ' he knows that Alice gets a sending receipt anyway, and that the Post Office will obtain a receipt from him (or issue a suitable affidavit) anyway.

Moreover, it can be arranged that eligible recipients in the ICM systems can incur additional charges if alternative actions to obtain a receipt from them are taken.

In the method just described, Bob is required to produce a receipt after he learns Alice's message, and her identifier if so wanted. The return-receipt method below, instead, elicits a receipt from Bob before he knows the message or the sender's identity. Nonetheless, the new receipt may still be used, if desired, to prove to third parties the content of Alice's message. In describing the preferred embodiment of the new return-receipt method, the same computational framework of the Sending-Receipt Method is assumed. In fact, the first step is identical to that of the Sending-Receipt Method.

THE RETURN-RECEIPT METHOD

A1 (Sender Step): Let m be the message that Alice wishes to send to Bob in a certified manner. Then she sends the Post Office an encrypted version of m intelligible by Bob but not by the Post Office.

Her transmission is preferably customized, signed, and indicates that it is part of an ICM transaction together with other valuable information, such as the transmission time. (e.g., she send $z=E_{PO}(SIG_A(ICM, B, T, E_B(m)))$.) PO1 (Post Office Step): The Post Office verifies who is the sender and who is the intended recipient, and

It sends Bob information that determines his message without making it yet intelligible to him.

In so doing the Post Office preferably hides Alice's identify, alerts Bob that he is dealing with an ICM transaction, and makes use of digital signatures. (e.g., it sends Bob $y=E_{PO}(SIG_{PO}(ICM, recipient: B, z))$ or ICM, $SIG_{PO}(E_B(B, z))$).

It also sends Alice a guarantee that it has done so.

Preferably, in so doing it also specifies other valuable information, such as time information T. (e.g., it sends Alice the value $x=E_A(SIG_{PO}(z, T))$.)

B1 (Recipient Step): Bob sends the Post Office a receipt that he got the above transmission. (e.g., he sends $E_{PO}(w)$, where $w=SIG_B(recipient, z)$).

Possibly, Bob's receipt also indicates other valuable information.

PO2 (Post Office Step): If Bob sends back the proper receipt within a specified amount of time, then the Post Office

1. sends Alice a suitable receipt; for instance, EA (w), and
2. sends Bob information that enables him to reconstruct Alice's message (e.g., $E_B(m)$).

If Alice has signed her transmission to the Post Office in Step A1 (e.g., she has sent the value z envisaged above), then it is preferable that the Post Office also enables Bob to guarantee the content of the message (e.g., it send Bob $SIG_A(ICM, B, T, E_B(m))$).

If Bob does not send back the proper receipt to the Post Office within a given amount of time, then the Post Office may either do nothing (in which case the only form of receipt in Alice's possession is what she has received from the Post Office in Step PO1); or inform Alice that it has received no receipt from Bob; or make a record that no receipt has been sent by Bob; or

PO3 takes action to deliver Alice's message to Bob in a way that is guaranteed to produce a return-receipt (e.g., it delivers the message to Bob by means of traditional certified mail). The thus obtained return receipt (or an affidavit that Bob refused willingly the mail) is then sent to Alice.

The above ICM transaction is a (logically) simultaneous one, and one that hides the identity of sender for as long as necessary.

The same variants and modifications for the Sending-Receipt Method can also be applied to the above method. Other variants may also be applied. In particular, the sending-receipt given by the Post Office to Alice in step PO1 may never be sent (e.g., because it may become irrelevant once Alice gets a return-receipt), or sent only if Bob does not produce a return-receipt fast enough. Also, the Post Office may receive a transmission from Alice before it performs its PO2 step. For instance, if Alice sends $E_AE_B(m)$ in Step A1, she is required to remove her encryption layer before Step PO).

If Bob receives the value z sent to him by the Post Office and properly acknowledges it (i.e., if all involved—including the. communication network—behave properly), the Return-Receipt Method is most efficient,

convenient and economical, since, in particular, it can be implemented in a pure electronic manner. In the Return-Receipt Method, Bob has even more incentives to produce his receipt than in the above modification of the Sending-Receipt Method. Indeed, for instance, while Alice may get a proper sending-receipt anyway that can prove the content of her message to him, if Bob refused to issue his better receipt, he will not even read the cleartext message, nor learn the sender's identity. Thus, while Alice already has a good form of receipt, by refusing to collaborate he has absolute nothing!

Despite the fact that Bob will almost always produce his receipts, the following are some practical ways to implement Step PO3. Here, the Post Office aims at delivering m to Bob in exchange for a receipt. Because the Post Office will not in general know m, it suffices that it delivers $E_B(m)$, or a string encompassing it. Without intending any restrictions, assume that the Post Office aims in Step PO3 at delivering the value $z=E_{PO}(SIG_{PO}(ICM, A, B, T, E_B(m)))$, envisaged in Step A1 and sent in digital form via a computer network.

To begin with, as discussed the delivery of z may occur by some version of traditional certified mail. For instance, the Post Office may print z on paper and then traditionally certified-mail deliver it to Bob, via a "mailman" which may or may not work for the Post Office (e.g., he may belong to UPS, Federal Express or other agency). The return-receipt obtained this way does not guarantee the content of the message, however, it may guarantee it in an indirect, yet adequate, way. For instance, it can be used in conjunction with a proper receipt of the Post Office (e.g., a digital signature of z sent to Alice in Step PO1) to provide evidence of the message actually delivered to Bob.

This format of z may be inconvenient, and thus create an extra incentive for Bob to issue a receipt in Step B1. Nonetheless, even this format of z may enable Bob to recover m: for instance, he may scan it (with character recognition) and then to put it into digital form prior to decrypting.

More conveniently, the Post Office may store z in a computer diskette and have it delivered in person to Bob. This form of delivery enables Bob to produce a return-receipt that guarantees directly the content. Indeed, upon being physically given the diskette, Bob may easily retrieve z from it and digitally sign it. This signature may then be given back to the mailman in the same diskette or in a different diskette. The mailman may indeed carry with him a device capable of checking Bob's signature. (This is quite feasible also because for signature checking such a device needs not to have access to any special secret).

Since Bob would be reading the message prior to signing it, it may be preferable to elicit first from Bob an ordinary generic receipt prior to giving him the diskette (in any case, the mailman can sign an affidavit that Bob accepted the diskette).

Alternatively, the diskette may contain not z, from which Bob may retrieve easily Alice's message, but information that pins down the message but does not yet reveal the message to Bob. For instance, the same value $y=E_{PO}(SIG_{PO}$ (ICM, recipient: B, z)) that we have envisaged the Post Office to send Bob in Step PO1. Only after Bob digitally signs y will the mailman enable Bob to retrieve Alice's message. For instance, the device carried by the mailman (preferably in a tamper-proof portion) may release a secret key by which Bob can remove the Post Office encryption layer. Alternatively, this key (or the right decryption, or information sufficient to decrypt anyway) can be sent, upon a proper signal, to the mailman, his device, or Bob directly by a variety of means (e.g., by phone, radio, etc.).

It should be understood that the present invention can be used to achieve additional properties, so as to yield other electronic transactions or make simultaneous other electronic transactions. For instance, the present ICM methods may be used to simultaneously sign contracts.

As for another example, it should also be appreciated that the ICM methods also yield very effective auctions methods with many bidding procedures (e.g., "public" or "secret" biddings). Indeed, Alice may be a bidder, Bob an entity handling the bids (e.g., deciding who are the winners of the auction, what goods are sold for what prices, how many units of a given good should be assigned to each bidder, and so on), and the message m for Alice to Bob is Alice's bid. Alice wishes to place her bid in return of a proper receipt, preferably one that can be used to prove (among other information, such as time information) the exact value of her bid. This way, if necessary, she can contest the "victory" of someone else. By means of our envisaged mechanisms for ICMs (in particular, of time information, encryption, and signatures), we can implement auctions in many different ways. Without any limitation intended, let us illustrate two possible implementations of two simple-minded auctions: one where the bidding process is "public" and one where it is "secret."

Consider first the following example of public bidding (which may occur, for instance, in a computer network). Assume there is a single indivisible good for sale in the auction, which will be assigned by a process combining both price and time. For making things cleaner, let us assume that there is a sequence of times $T_1, T_2, \ldots$ and $T'_1, T'_2, \ldots$ where $T_i \leqq T'_i$ (e.g. $T'_i = T_i + \Delta$, where Δ is a fixed quantity.) A bidder gets the goods for a price P if there is an index/such that she has offered a price P within time T_i and no higher price has been offered by time T'_i. (It is thus advisable that T'_i be greater than T_i, so that there is sufficient time to process all bids properly.)

The current status of the bid can be made available (e.g., by Bob), so that the bidders know what the highest offered price, P, at the "current" time, T, is. If Alice is willing to raise the price, she must do so before it is too late. Since her bid consists of her message to Bob, and it is assumed that the Sending-Receipt Method is in use, Alice then sends here bid to the Post Office in Step A1. If this transmission arrives within a useful time (i.e., before some time T'), the Post Office issues her a receipt with an indication of the proper time (interval), and then forwards her bid to Bob. Bob then processes the bids relative to the next time interval (e.g. announces the new highest price, or that the auction is over because no one offered more than the previous highest price).

As can be seen, the Post office may in this application be an entity cooperating with Bob, even for only auction purposes. Nonetheless, it may be preferable that it be made sufficiently independent from Bob. For instance, though prices are meant to be public, it is useful that bids are encrypted with Bob's key, so that the Post office will not know the content of a bid when it issues a receipt. Thus, in particular, it cannot be blamed to have refused to issue a receipt (e.g., by claiming that it had arrived too late) in order to favor a particular bidder. On the other hand, Bob, though capable to read the bids, is held back from cheating by the fact that the bidders have been issued valid and very informative receipts.

The system can be further enhanced so that the identity of the bidder is not revealed to Bob (at least as long as the auction is going on), but, say, only the price and time information. Also, at each time (interval), rather than mak-

ing available just the new highest bid/price, Bob may make available all incoming (legitimate) bids, so that the volume of bidding is also learned by the bidders. Also, rather than processing the incoming bids in batches and in time intervals, Bob may process them one at a time (preferably in the order they got in) and with individual times. (e.g., he may still announce only the currently highest bid with its own individual time T, and when a bid with price P and time T is announced, and no higher price than P is offered before time T+Δ then the auction is over.) Again, return receipt may also be used in this application.

It should also be noted that if Alice has sent her bid in a very timely fashion and has not received any timely receipt within a certain time, then she may still time to take additional steps to ensure that her bid is properly delivered. Again, having two or more Post Offices, or Post Offices comprising a plurality of agents, may be very useful here because this enhance her chance of getting at least one valid receipt.

In particular the Post Office agents may be implementing a threshold cryptosystem. A plurality of Post Offices or multi-agent Post Offices may also benefit Bob, because he is better guaranteed that each bid will be properly forwarded to him. There may also be more than one Bob, and (each) Bob too may comprise several agents. It should be appreciated that if there are a multiplicity of agents involved it is also possible that Bob and the Post Office coincide, that is, that they simply are names for different functions performed by the same auctioning entity.

Notice also that the ICM methods may immediately accommodate secret bidding mechanism. Indeed, any of the methods above may be used for this purpose. For instance, consider batch-processing of bids when there is a single time interval Tand a single, disjoint and subsequent time interval T'. Then the Post Offices issues receipts only for those bids received during T, and forwards all these bids to Bob, but only during T'. This way, no bid can be learned before the right time, unless there is an illegitimate cooperation between Bob and the Post Office (or sufficiently many agents). In all these scenarios, customization is quite useful since it also prevents that an enemy can copy Alice's bid so as to be guaranteed that he will win the auction if she does.

Finally, it should be noticed that the methods extend to more complex auctions, (e.g., there may be may goods of arbitrary nature—such as airwave bandwidths—, these goods may be divisible, and thus, for instance, the highest bid may take only a portion of a good, and so on.) In general it will be important to also indicate in each bid the particular, auction, good, and the like.

Although the invention has been described in detail, it should be appreciated that the scope of the invention is limited only, by the following claims.

What is claimed is:

1. A method of transmitting a message using a trusted party, comprising:

a sender causing a customized version of the message to be provided to the trusted party, the customized version of the message having a first portion intelligible to the trusted party but not to a recipient of the message and a second portion intelligible to the recipient of the message but not to the trusted party;

the trusted party examining the first portion of the customized version of the message to determine the recipient;

the trusted party causing at least the second portion of the customized version of the message to be provided to the recipient; and

the trusted party causing a receipt for the message to be provided to the sender.

2. An electronic communication method comprising:

sending from a first party a message for a trusted party, the message having first and second portions, the first portion being intelligible to the trusted party, identifying a second party as a recipient of the second portion and being unintelligible to the second party, the second portion being unintelligible to the trusted party and intelligible to the second party; and

receiving by the first party a receipt indicating that the second portion of the message was received by the second party.

3. The method of claim 2 wherein the first portion of the message is information encrypted to render it unintelligible to the second party.

4. The method of claim 2 wherein the second portion of the message is information encrypted to render it unintelligible to the trusted party.

5. The method of claim 2 further comprising signing at least one of the first and second portions of the message by the first party.

6. The method of claim 2 wherein the receipt includes a representation of the second portion of the message.

7. The method of claim 2 wherein the receipt includes a signature of at least the recipient.

8. The method of claim 2 wherein:

the second portion includes information which has been processed to render it unintelligible to the trusted party and intelligible to the second party; and

the second portion can be reconstructed using the information.

9. The method of claim 2 wherein an identity of the second party is intelligible from the message only by the trusted party.

10. The method of claim 2 wherein an identity of the second party is intelligible from the receipt only by the first party.

11. An electronic communication method comprising:

receiving by a trusted party a message from a first party, the message having first and second portions, the first portion being intelligible to the trusted party, identifying a second party as a recipient of the second portion and being unintelligible to the second party, the second portion being unintelligible to the trusted party and intelligible to the second party;

sending by the trusted party the second portion of the message to the second party; and

sending by the trusted party to the first party a receipt indicating that the message was delivered to the second party.

12. The method of claim 11 wherein the first portion of the message is information encrypted to render it unintelligible to the second party.

13. The method of claim 11 wherein the second portion of the message is information encrypted to render it unintelligible to the trusted party.

14. The method of claim 11, wherein the message includes the first party's signature of at least one of the first and second portions of the message.

15. The method of claim 11 wherein:

the second portion is information which has been processed to render it unintelligible to the trusted party and intelligible to the second party; and

the message can be reconstructed using the information.

16. The method of claim 11 wherein an identity of the second party is intelligible from the message only by the trusted party.

17. The method of claim 11 wherein an identity of the second party is intelligible from the receipt only by the first party.

18. The method of claim 11 wherein sending the second portion of the message to the second party includes signing at least the second portion of the message with the trusted party's signature.

19. The method of claim 18 wherein sending the second portion of the message to the second party further comprises processing the signed second portion to render it intelligible to the second party but unintelligible to at least one party other than the second party.

20. The method of claim 11 wherein the receipt includes a representation of the second portion of the message.

21. The method of claim 11 wherein the receipt includes a signature of at least the recipient.

22. The method of claim 11 wherein sending the second portion of the message to the second party by the trusted party comprises:

generating by the trusted party a processed message which determines the second message but which is unintelligible to the second party;

sending the processed message to the second party by the trusted party;

receiving by the trusted party a receipt indicating that the second party received the processed message; and

sending the second message to the second party in a form intelligible to the second party.

23. The method of claim 22 wherein the processed message can be reconstructed from the second message.

24. The method of claim 22 wherein the receipt indicating that the second party received the processed message includes a signature of the second party.

25. An electronic communication method comprising:

receiving by a receiver a first message from a trusted party, the message having a portion normally intelligible to the receiver which has been processed to render it unintelligible to the receiver;

sending by the receiver a receipt for the message to the trusted party; and

receiving by the receiver a second message from the trusted party, the second message including the portion intelligible to the receiver.

26. The method of claim 25, wherein the receipt can be reconstructed using the portion of the second message intelligible to the receiver.

27. The method of claim 25, wherein the first message is intelligible to the trusted party.

28. The method of claim 25, wherein sending the receipt includes signing the first message by the receiver.

29. The method of claim 28, wherein sending the receipt further includes processing the receipt to render it intelligible to the trusted party but unintelligible to at least one other party.

* * * * *

US006154543A

United States Patent [19]

Baltzley

[11] Patent Number: 6,154,543

[45] Date of Patent: Nov. 28, 2000

[54] **PUBLIC KEY CRYPTOSYSTEM WITH ROAMING USER CAPABILITY**

[75] Inventor: **Cliff A. Baltzley**, Austin, Tex.

[73] Assignee: **Hush Communications Anguilla, Inc.**

[21] Appl. No.: **09/200,640**

[22] Filed: **Nov. 25, 1998**

[51] **Int. Cl.**[7] **G09C 3/08**

[52] **U.S. Cl.** **380/255**; 380/259; 705/74; 713/150; 713/201

[58] **Field of Search** 380/255, 259; 705/64, 74, 75; 713/150

[56] **References Cited**

U.S. PATENT DOCUMENTS

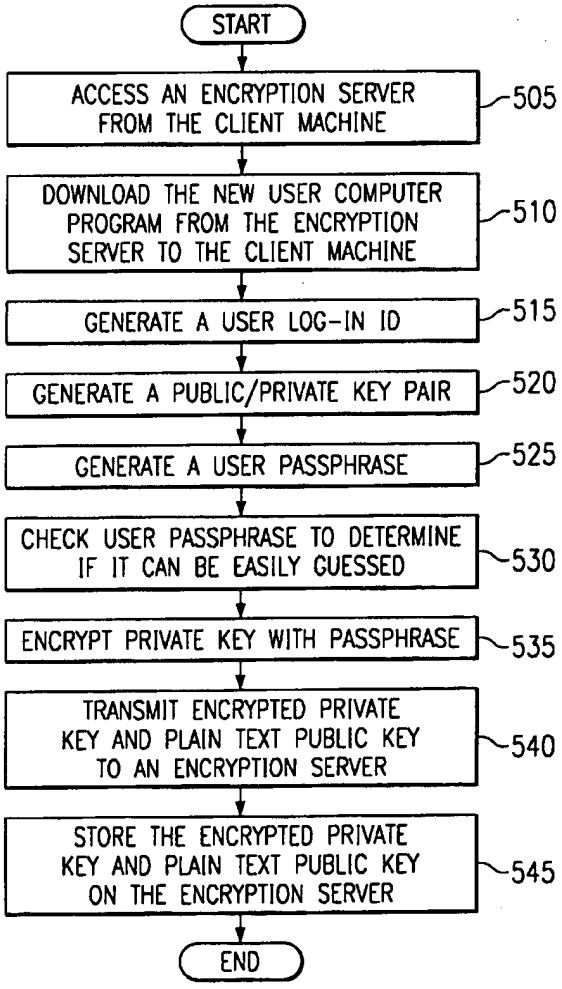
4,200,770	4/1980	Hellman et al.	178/22
4,405,829	9/1983	Rivest et al.	178/22.1
5,619,574	4/1997	Johnson et al.	380/25
5,748,735	5/1998	Ganesan	380/21
5,757,916	5/1998	MacDoran et al.	380/25
5,903,652	5/1999	Mital	380/25
5,987,440	11/1999	O'Neil et al.	705/44

Primary Examiner—Thomas R. Peeso
Assistant Examiner—Todd Jack
Attorney, Agent, or Firm—Gray Cary Ware & Freidenrich LLP

[57] **ABSTRACT**

A public key cryptosystem with roaming user capability within a network that allows secure communication between users of the system, client machines, and encryption servers. A client machine generates and stores an encrypted private key on an encryption server. A user may then access the encrypted private key from any client machine located on the network and decrypt it using a passphrase, thus giving the user roaming capability. The private key may then be used to decrypt any encrypted messages received. A user can generate a digital message, encrypt it with a client recipient's public key, and transmit it to the encryption server from any client machine on the network.

59 Claims, 5 Drawing Sheets



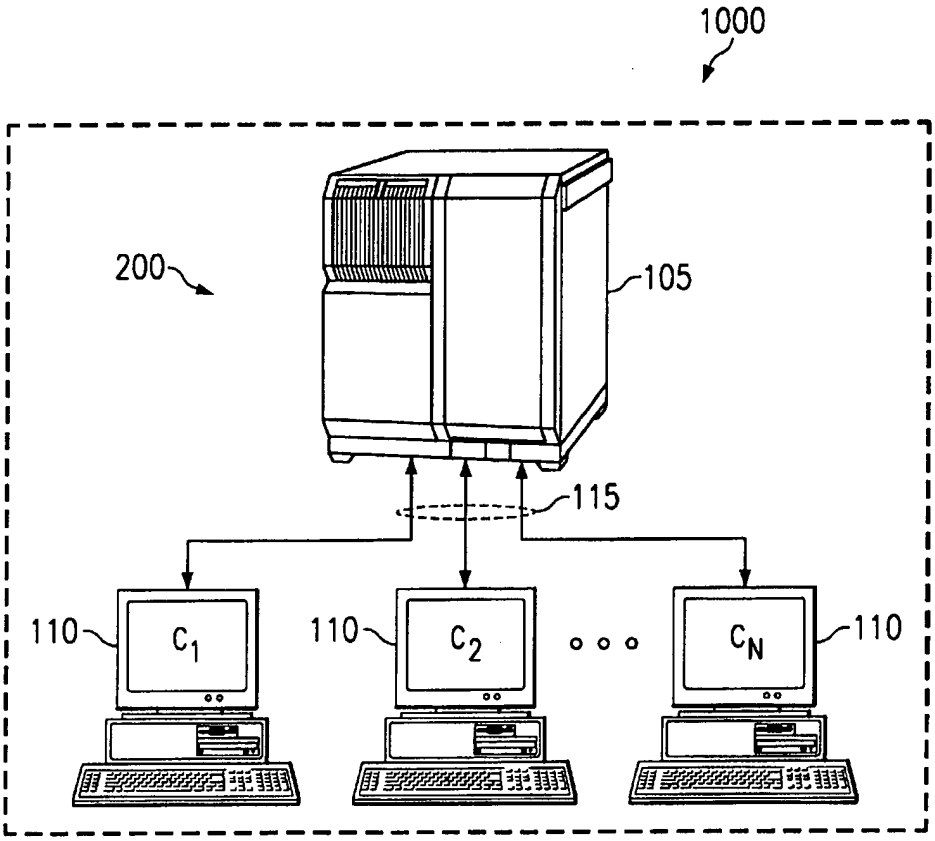

1000
200
105
115
110
C_1
110
C_2
C_N
110

FIG. 1

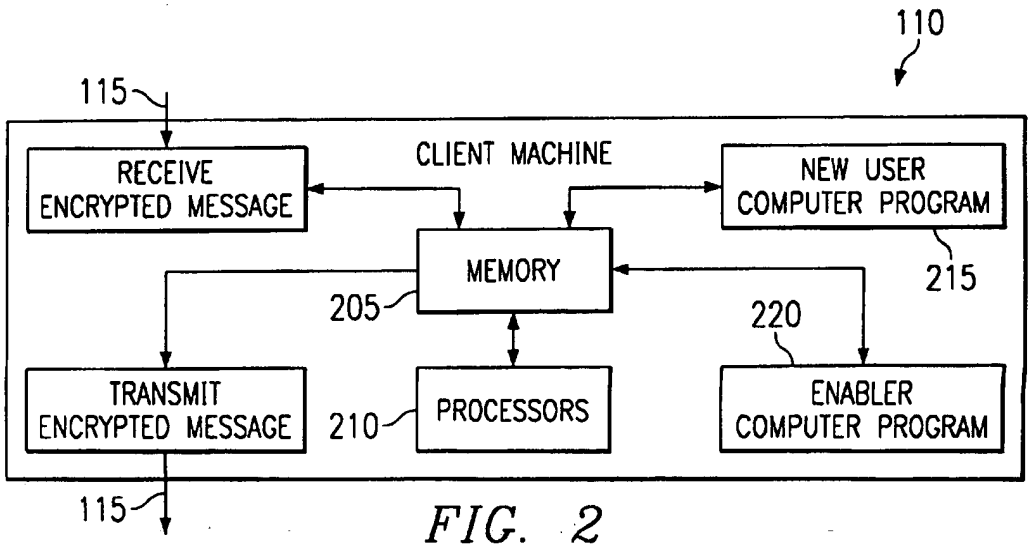

110
115
CLIENT MACHINE
RECEIVE ENCRYPTED MESSAGE
NEW USER COMPUTER PROGRAM
215
MEMORY
205
220
TRANSMIT ENCRYPTED MESSAGE
210
PROCESSORS
ENABLER COMPUTER PROGRAM
115

FIG. 2

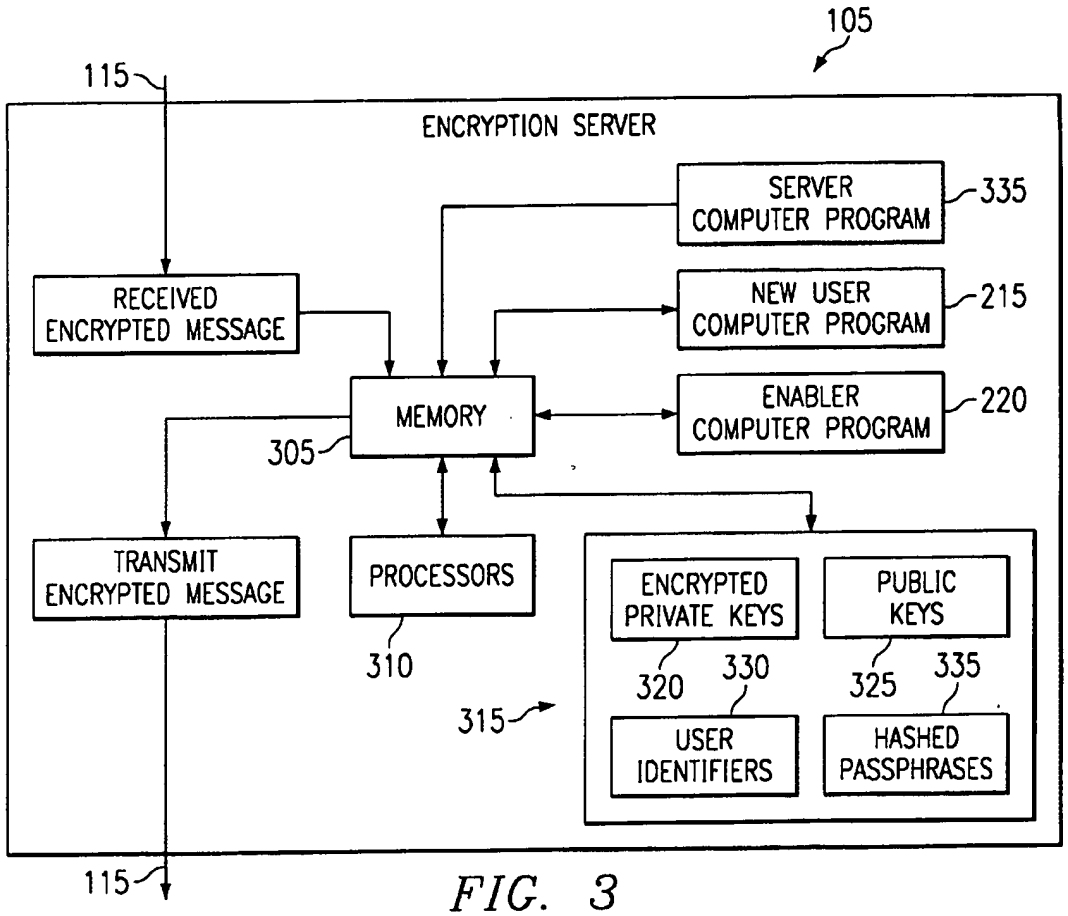

105
115
ENCRYPTION SERVER
SERVER COMPUTER PROGRAM
335
RECEIVED ENCRYPTED MESSAGE
NEW USER COMPUTER PROGRAM
215
MEMORY
305
ENABLER COMPUTER PROGRAM
220
TRANSMIT ENCRYPTED MESSAGE
PROCESSORS
310
ENCRYPTED PRIVATE KEYS
PUBLIC KEYS
320
330
325
335
315
USER IDENTIFIERS
HASHED PASSPHRASES
115

FIG. 3

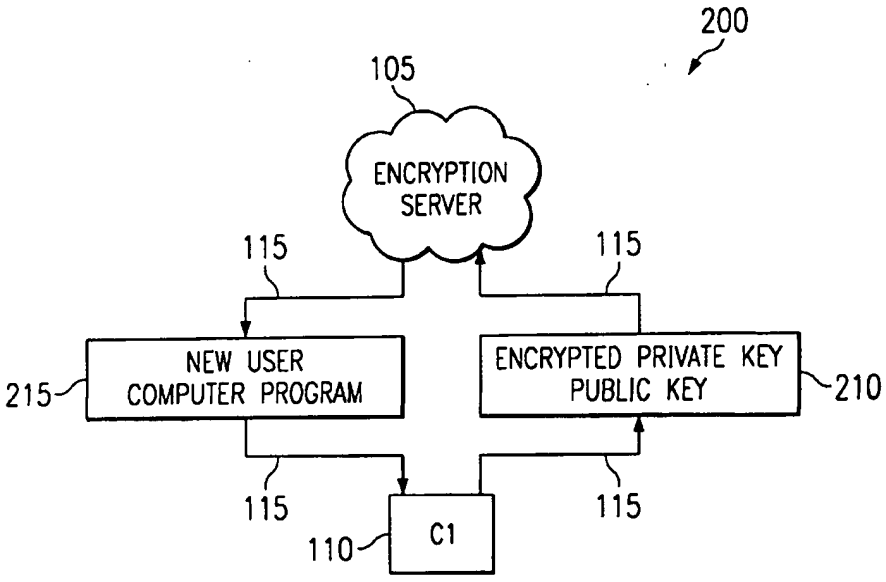

200
105
ENCRYPTION SERVER
115
115
NEW USER COMPUTER PROGRAM
215
ENCRYPTED PRIVATE KEY PUBLIC KEY
210
115
115
110
C1

FIG. 4

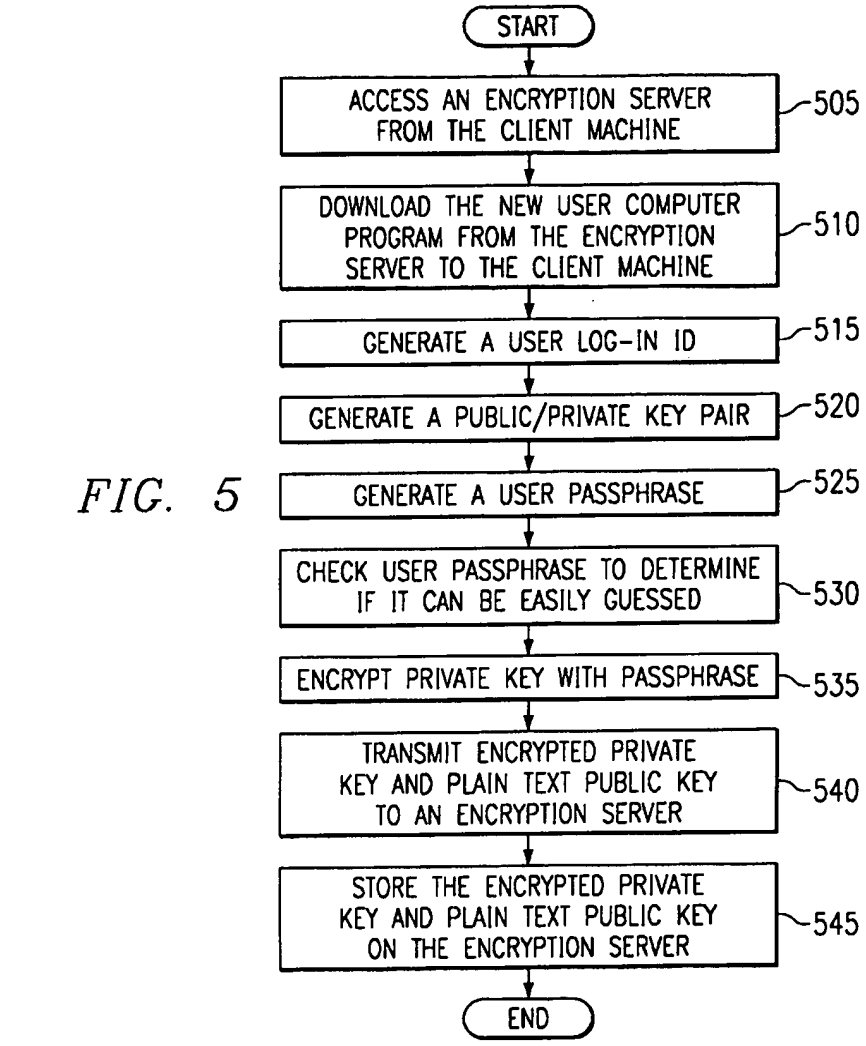

START
ACCESS AN ENCRYPTION SERVER FROM THE CLIENT MACHINE
505
DOWNLOAD THE NEW USER COMPUTER PROGRAM FROM THE ENCRYPTION SERVER TO THE CLIENT MACHINE
510
GENERATE A USER LOG-IN ID
515
GENERATE A PUBLIC/PRIVATE KEY PAIR
520
GENERATE A USER PASSPHRASE
525
CHECK USER PASSPHRASE TO DETERMINE IF IT CAN BE EASILY GUESSED
530
ENCRYPT PRIVATE KEY WITH PASSPHRASE
535
TRANSMIT ENCRYPTED PRIVATE KEY AND PLAIN TEXT PUBLIC KEY TO AN ENCRYPTION SERVER
540
STORE THE ENCRYPTED PRIVATE KEY AND PLAIN TEXT PUBLIC KEY ON THE ENCRYPTION SERVER
545
END

FIG. 5

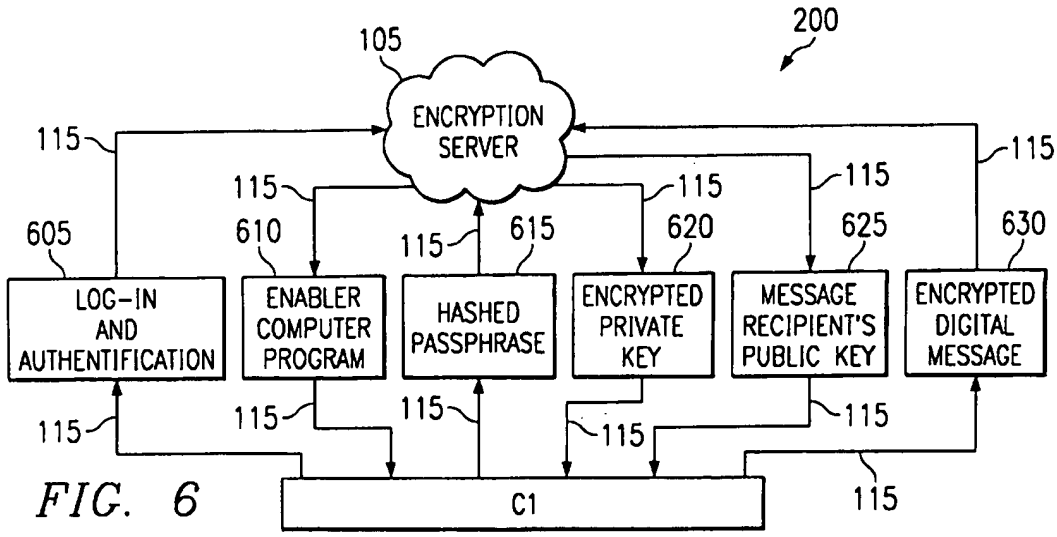

200
105
ENCRYPTION SERVER
115
605
LOG-IN AND AUTHENTIFICATION
610
ENABLER COMPUTER PROGRAM
615
HASHED PASSPHRASE
620
ENCRYPTED PRIVATE KEY
625
MESSAGE RECIPIENT'S PUBLIC KEY
630
ENCRYPTED DIGITAL MESSAGE
C1

FIG. 6

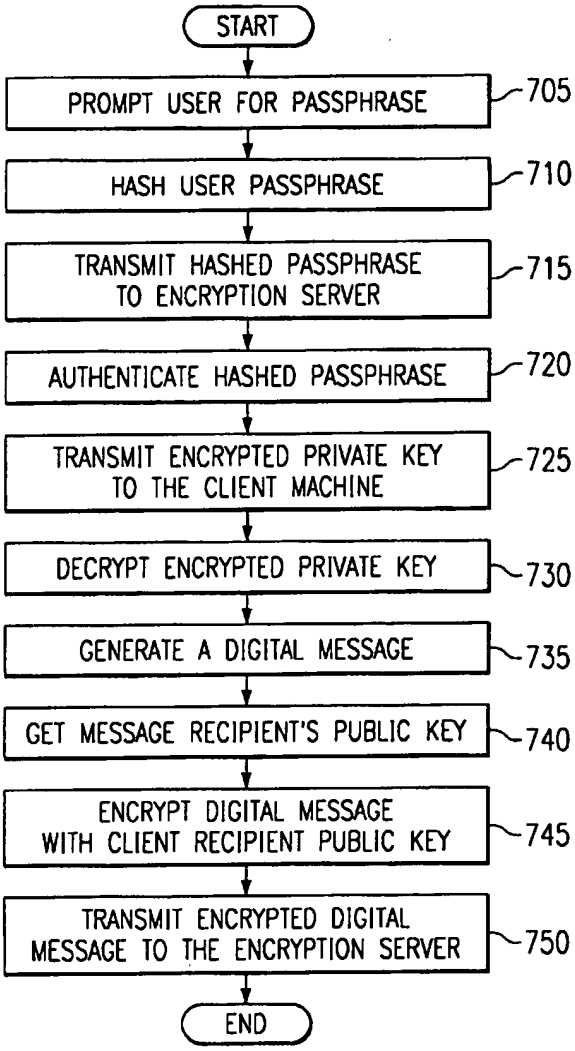

START
PROMPT USER FOR PASSPHRASE
705
HASH USER PASSPHRASE
710
TRANSMIT HASHED PASSPHRASE TO ENCRYPTION SERVER
715
AUTHENTICATE HASHED PASSPHRASE
720
TRANSMIT ENCRYPTED PRIVATE KEY TO THE CLIENT MACHINE
725
DECRYPT ENCRYPTED PRIVATE KEY
730
GENERATE A DIGITAL MESSAGE
735
GET MESSAGE RECIPIENT'S PUBLIC KEY
740
ENCRYPT DIGITAL MESSAGE WITH CLIENT RECIPIENT PUBLIC KEY
745
TRANSMIT ENCRYPTED DIGITAL MESSAGE TO THE ENCRYPTION SERVER
750
END

FIG. 7

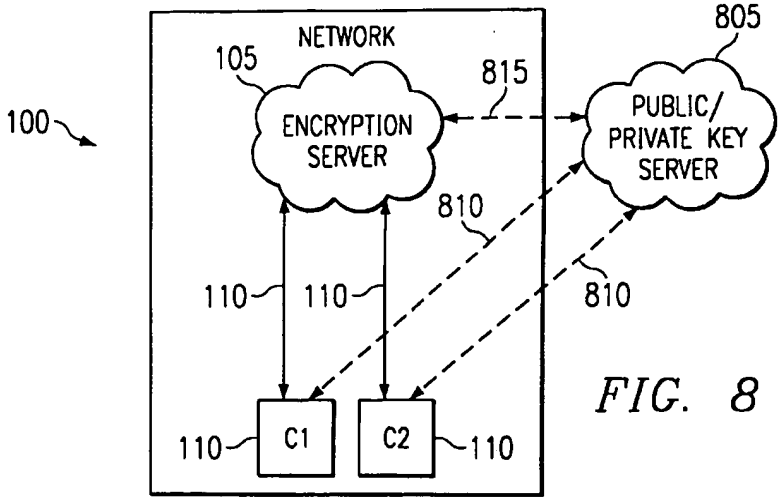

NETWORK
105
815
805
100
ENCRYPTION SERVER
PUBLIC/ PRIVATE KEY SERVER
810
110
110
810
110
C1
C2
110

FIG. 8

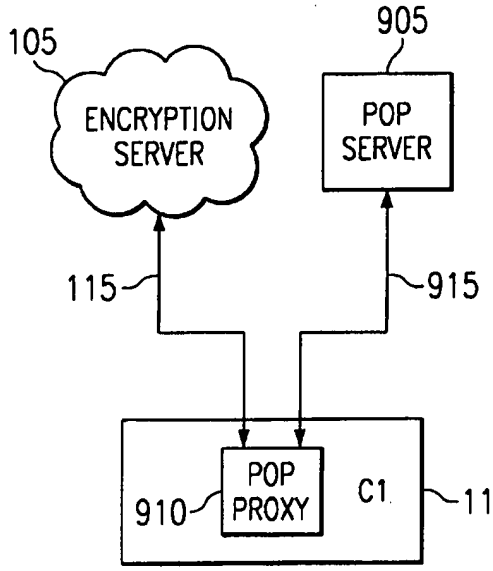

105
ENCRYPTION SERVER
905
POP SERVER
115
915
910
POP PROXY
C1
110

FIG. 9

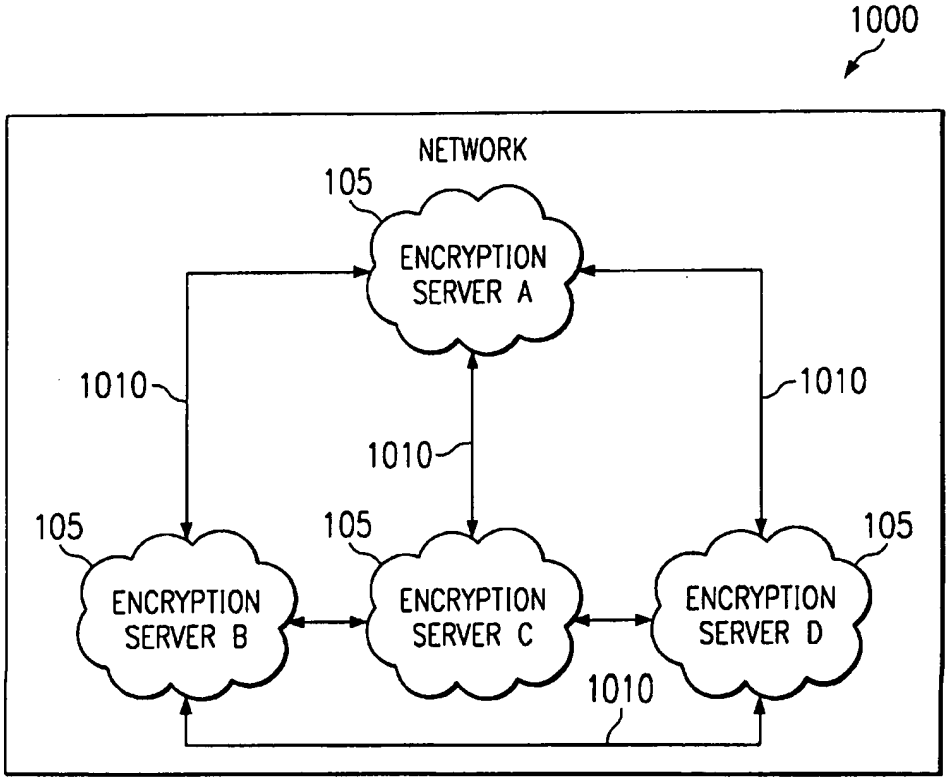

1000
NETWORK
105
ENCRYPTION SERVER A
1010
1010
1010
105
105
105
ENCRYPTION SERVER B
ENCRYPTION SERVER C
ENCRYPTION SERVER D
1010

FIG. 10

PUBLIC KEY CRYPTOSYSTEM WITH ROAMING USER CAPABILITY

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to encryption of data in communication systems. In particular, this invention relates to a system and method for managing public/private key pairs within a cryptosystem having roaming user capability.

BACKGROUND OF THE INVENTION

Encrypted voice and data communication systems are well known in the art. These cryptosystems allow a user to digitally transmit information to one or more system users without it being intercepted and interpreted. This is accomplished by encrypting and decrypting the transmitted information with what is known as an encryption key. Encryption keys may be secret keys, where a single key is utilized for encryption and decryption, or public keys, where two or more keys are used.

Cryptosystems which utilize secret keys and public keys are well known in the art. Each type of cryptosystem provides some degree of privacy and authentication for digital communications. Secret-key cryptosystems utilize the traditional method known as symmetric key cryptography. In a symmetric key cryptosystem, a single electronic key is used both to encrypt and decrypt the transmitted information. Since only one key is used, the sender must provide the receiver with the key by some form of secure communication. The lack of a secure channel, which is usually why encryption is used in the first place, makes this system mostly obsolete in common practice these days.

Public-key cryptosystems, also referred to as asymmetric cryptosystems, provide another means of encrypting information. Such cryptosystems differ from secret-key cryptosystems in that two or more keys are required as opposed to one. In a public-key cryptosystem, each entity has a private key and a public key. Public keys are generally held in databases run by "Key Certificate Authorities" and are publicly known. However, each user's private key is known only by that user. Once a sender encrypts a message with a recipient's public key, it can only be decrypted using that recipient's private key. Because the computational power required to break a key increases exponentially with the length of key, longer keys provide greater security.

Private keys are usually between 512 and 4096 bits long, far too long for the average person to commit to memory. For this reason, most users of a public key cryptosystem store their private key on a personal computer or other personal device. The problem with this practice is that private key may be lost if the computer software crashes or computer hardware fails. In most cases, the user may have not "backed up" their data. This situation occurs more often than is convenient. In the event that the user wrote down the private key in a "safe" place and then lost it, the result is the same.

If or when this private key is lost or stolen, and thus compromised, a complicated "Key Revocation" process occurs. The user must perform the embarrassing task of informing all other users with whom he or she communicates with that the public/private key pair is no longer valid, and provide them with a new public key to use instead.

Another major drawback with current public key cryptosystems is that the users must have their private key with them to read any of their messages. This becomes a problem when the user is traveling and the private key is stored on their personal computer at home. In the current age of "roaming email" and other roaming communication, the technology is readily available for users to check their messages almost anywhere in the world. If the users do not have their private key with them, they cannot retrieve their messages. If the users do carry their private key with them while traveling, there is the risk that the private key may be lost or stolen. Furthermore, it is not always easy or convenient for users to carry around a piece of digital data with them that quickly integrates with other digital hardware worldwide.

SUMMARY OF THE INVENTION

The present invention provides a system and method for transmitting secure digital electronic messages over communication channels in a way that substantially eliminates or reduces disadvantages and problems associated with previously developed cryptosystems.

More specifically, the present invention provides a system and method for providing a public key cryptosystem having roaming user capability. The public key cryptosystem with roaming user capability comprises a network having multiple client computers and multiple encryption servers. The network allows secure communication between the client computers and the encryption servers.

In one embodiment, the client computer executes a New User computer program and an Enabler computer program to facilitate secure communication. Both the New User computer program and the Enabler computer program communicate with a Server computer program located on the encryption server. The New User computer program communicates with the Server computer program to generate a public/private key pair, a user identifier, and a user passphrase. The private key is then encrypted with the user passphrase, yielding an encrypted private key, which is transmitted with the public key to the encryption server.

The Enabler computer program communicates with the Server computer program to enable a user to both read encrypted digital messages sent to him or her and send encrypted digital messages to other users. To read encrypted digital messages sent to a user, the user is first prompted for a passphrase. The passphrase is then hashed and transmitted to the encryption server for authentication. Once the hashed passphrase is authenticated, the encryption server transmits the user's encrypted private key back to the client computer, where it is decrypted. The user may now use the private key to read any digital messages he has received.

The Enabler computer program and the Server computer program also work in conjunction to send encrypted digital messages. Once a digital message is generated, it is encrypted with a client recipient's public key. The encrypted message is then transmitted to the client recipient computer.

The present invention provides an important technical advantage by providing a way to securely store a user's private key on an encryption server by symmetrically encrypting it with a passphrase so that no one but the user has access to it.

The present invention provides another important technical advantage by providing a way to securely store a user's private key on an encryption server so a user may access the private key from any client machine on the encryption server network, thus providing roaming capability.

The present invention provides another important technical advantage by providing a way to access an encrypted private key from any client machine on a network by simply remembering a user passphrase.

The present invention provides another important technical advantage by providing a way to store an encrypted private key on an encryption server instead of the user's client machine, thus preventing the loss of the private key in the event the client machine crashes or fails.

The present invention provides another important technical advantage by limiting the number of times a user may try to log-in to the network per hour so a hacker cannot break into the system and retrieve the user's encrypted private key.

The present invention provides another important technical advantage by providing a user friendly public key cryptosystem where the user need not understand how to generate, send, or receive a public/private key pair since all this is handled by the New User computer program, Enabler computer program and the Server computer program.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numerals indicate like features and wherein:

FIG. **1** shows one embodiment of a communications network system comprising an encryption server, multiple client machines, multiple users, and communication channels in accordance with the invention;

FIG. **2** shows a diagram of a client machine comprising incoming and outgoing communication channels, a New User computer program, an Enabler computer program, memory, and processors;

FIG. **3** shows a diagram of an encryption server comprising incoming and outgoing communication channels, a New User computer program, an Enabler computer program, a Server computer program, memory, processors, and a database having a plurality of encrypted private keys, public keys, user identifiers and hashed passphrases;

FIG. **4** shows a system diagram of an encryption server downloading a New User computer program, running the New User computer program on a client machine, and transmitting an encrypted private key and public key back to the encryption server in accordance with this invention;

FIG. **5** shows a flow chart detailing the functions performed by the New User computer program in accordance with this invention;

FIG. **6** shows a system diagram of the process of logging-in to the encryption server from a client machine, downloading the Enabler computer program to the client machine, transmitting a hashed passphrase to the encryption server, downloading the encrypted private key, downloading a client recipient's public key, and generating and sending an encrypted digital message to the encryption server in accordance with the invention;

FIG. **7** shows a flow chart detailing the functions performed by the Enabler computer program in accordance with this invention;

FIG. **8** shows another embodiment of a communications network comprising an encryption server, a plurality of client machines, and a public/private key server located outside the communications network.

FIG. **9** shows another embodiment of a communications network comprising an encryption server, a pop server, and a client machine containing a pop proxy; and

FIG. **10** shows a network comprising multiple encryption servers all connected to each other through communication channels.

DETAILED DESCRIPTION OF THE INVENTION

Preferred embodiments of the present invention are illustrated in the FIGURES, like numerals being used to refer to like and corresponding parts of the various drawings.

FIG. **1** shows one embodiment of the public key cryptosystem with roaming user capability **200** of the present invention within a communication network system **1000** comprising an encryption server **105** connected to a network of multiple client machines **110** through communication channels **115** which may each be comprised of a secure socket layer. The public cryptosystem with roaming user capability **200** may have a firewall or any other security devices placed between the encryption server **105** and the client machines **110** to further secure the encryption server **105** from being hacked or broken into.

FIG. **2** shows a client machine **110** which can comprise incoming and outgoing communication channels **115**, a memory **205**, and one or more processors **210**, such as microprocessors or digital signal processors. Memory **205** can include any storage medium, including RAM, a hard drive, and tape memory. The processors **210** are electrically connected to the memory **205** and have access to a New User computer program **215** and an Enabler computer program **220**. The New user computer program **215** and Enabler computer program **220** may be downloaded from the encryption server **105** and stored in memory **205** of client machine **110** or directly installed in the memory **205** of client machine **110** from some other source. Both the New User computer program **215** and Enabler computer program **220** communicate with a Server computer program located in memory **305** of the encryption server **105**. One example of a client machine **110** is an IBM compatible computer, however, it should be understood that the client machine **110** can be any communication unit which contains input and output communication channels **115**, memory **205**, and processors **210**.

FIG. **3** shows an encryption server **105** which may comprise input and output communication channels **115**, a memory **305**, a database **315**, and one or more processors **310**, such as microprocessors or digital signal processors. The database **315** may comprise a plurality of encrypted private keys **320**, a plurality of public keys **325**, a plurality of user identifiers **330** and a plurality of hashed passphrases **335**. The user identifiers could be a log-in ID, or a passphrase. The processors **310** are electrically connected to the memory **305** and have access to a Server computer program **335**. The Server computer program **335** may be divided into two or more subprograms. The New User computer program **215** and an Enabler computer program **220** may be resident on the encryption server **105** and accessible by the client machines **110**. One example of an encryption server **105** is a Sun Spare Station **5**, however, it should be understood that the encryption server **105** can be any communication unit which contains input and output communication channels **115**, memory **305**, and processors **310**.

FIG. **4** shows one embodiment of the public key cryptosystem with roaming user capability **200** where a user may access a web page on the client machine **110** and download the New User computer program **215** to the client machine **110** from the encryption server **105**. The New User computer program **215** may also be downloaded from a server outside the network **1000** or directly loaded on to the client machine **110** from another source. The New user computer program **215** directs the client machine **110** to generate a user identifier **330**, a private key, and a public key **325**. The New

User computer program **215** then encrypts the private key and transmits the encrypted private key **320** and public key **325** back to the encryption server **105**. The Server computer program **335** directs the encryption server **105** to receive the encrypted private key **320** and the public key **325** from the client machine **110** and store them in the encryption server **105** database **315**.

FIG. **5** shows the steps performed by one embodiment of the New User computer program **215** working in conjunction with the Server computer program **335**. The user first accesses an encryption server **105** from the client machine **110** as stated in step **505**. The encryption server **105** may be accessed from the client machine **110** through an encryption server **105** web page. The user then downloads the New User computer program **215** from the encryption server **105** to the client machine **110** in step **510**. At steps **515**, **520** and **525** respectively, the New User computer program **215**, which may be written in a number of different computer languages including JAVA, generates a user identifier **330**, private key, public key **325**, and prompts the user for a user passphrase. The user may choose his own passphrase or let the New user computer program **215** generate it for him. True random numbers needed to facilitate key generation may be actively or passively generated by the user during this time. The New user computer program **215** then communicates with the Server computer program **335** and compares the hash of the user passphrase against a large database of hashed English words, hashed common nouns, and hashed popular sayings to assure that the hash of the passphrase chosen cannot be easily guessed in step **530**. If the passphrase is determined to be easily guessable, the user has the option to either keep the passphrase or generate a new one. The New User computer program **215** then encrypts the private key with the passphrase in step **535**. The private key may be encrypted with a number of different ciphers, including a symmetrical cipher such as Blowfish or DES. In step **540**, the encrypted private key **320** and public key **325** are then transmitted to the encryption server **105**. Finally, the Server computer program **335** stores the encrypted private key **320** and public key **325** on the encryption server **105** in step **545**. In another embodiment, the New User computer program **215**, the Enabler computer program **220**, the encrypted private key **320**, and other user preference information may be stored on the client machine **110** as well as transmitting and storing it on the encryption server **105** to save download transmission time.

By storing the encrypted private key **320** on the encryption server **105**, the user enjoys some added benefits. First, the user may access and download the encrypted private key **320** from any client machine **110** on the network **1000**, thus giving the user roaming capability. Second, storing the encrypted private key **320** on the encryption server **105** eliminates the need for the user to remember or carry his or her private key. All the user needs to remember to access the encrypted private key **320** is a passphrase. This is considerably easier than remembering a private key which may be as large as 2,048 bits. Third, since the user's private key is stored on the encryption server **105** in encrypted form, only the user may retrieve and decrypt the private key. Neither an encryption server **105** administrator nor anyone else would be able to decrypt the private key.

FIG. **6** shows one embodiment of the public key cryptosystem with roaming user capability **200** depicting the process by which a client machine **110** transmits a digital message to the encryption server **105**. First, the user logs-in to the encryption server **105** in step **605**. Here, the server is authenticated to the user by industry standard means, such as SSL using authentication certificates. For security purposes, a user may be limited to a certain number of log-in sessions per hour, such as forty, to prevent someone from trying to break into the network **1000** and obtain a user's encrypted private key **320**. The encryption server **105** then downloads the Enabler computer program **220** to the client machine **110** in step **610**. The user then enters his or her passphrase, hashes the passphrase, and transmits the hashed passphrase to the encryption server **105** in step **615**. In step **620**, the encryption server **105** authenticates the hashed passphrase and transmits the encrypted private key **320** back to the client computer **110**. In step **625**, the user may decrypt the encrypted private key **320** with his or her passphrase, generate a digital message, and obtain a message recipient's public key **325** from the encryption server **105**. Finally, in step **630**, the user may encrypt the digital message with the recipient's public key **325**, optionally signing the digital message with the client sender's private key, and transmit the encrypted digital message to the encryption server **105**. All public keys **325** of message recipients may be temporarily or permanently stored on the client machine **110** for speed in future message sending.

Once the encrypted digital message is stored on the encryption server **105**, the client recipient to whom the encrypted digital message is directed may retrieve and decrypt the encrypted digital message with his private key. The digital message may be email, real-time chat, or any other form of digital message which may be transmitted over the network **1000**.

In another embodiment, the encrypted digital message does not have to be stored on the encryption server **105**, but may instead be transmitted in any convenient way to the digital message recipient. For real time data that is time or bandwidth sensitive, (e.g., real time voice communication) encrypted digital message data may flow directly between both communicating client machines **110**. The encryption server **105** is only necessary for user key storage.

In the process depicted in FIG. **6**, the user passphrase, plaint text private key, or encrypted private key **320** remains on the client machine **110** only for the duration of time in which the user is logged-in to the network **1000**. As soon as the user logs-off of the network **1000**, the passphrase is erased from the client machine **110**.

In another embodiment, the user passphrase, or private key may not be erased after logging-off the network **1000**. In this embodiment, the user passphrase or private key remain on the computer so the user rarely has to retype their passphrase or download the encrypted private key **320** from the encryption server **105**. The user passphrase or plain text private key, are never transmitted to the encryption server **105**.

FIG. **7** details the functions performed by one embodiment of the Enabler computer program **220** working in conjunction with the Server computer program **335**. In step **705**, the Enabler computer program **220** first prompts the user for a passphrase. The passphrase is then hashed and transmitted to the encryption server **105** in steps **710** and **715**. The Server computer program **335** authenticates the hashed passphrase and transmits the encrypted private key **320** back to the client machine **110** in steps **720** and **725**. The Server computer program **335** may also transmit other user information from the encryption server **105** to the client machine **110**. In step **730**, the Enabler computer program **220** then decrypts the encrypted private key **320** at the client machine **110**. At this point, the user may use his or her private key to access his or her digital messages.

The Enabler computer program **220** also allows the user to generate a digital message and obtain a recipient's public key **325** from the encryption server **105** as shown in step **735** and **740**. Finally, in steps **745** and **750**, the Enabler computer program **220** encrypts the digital message with a client recipient public key **325** and transmits the encrypted digital message to the encryption server **105**. A cyclic redundancy check (CRC) may be added to the end of the digital message before encrypting it for added security. A couple of examples of ciphers which may be used to encrypt the digital message are the standard RSA cipher or the Diffie-Helman cipher.

FIG. **8** shows another embodiment of the public key cryptosystem with roaming user capability **1000** where the client machines C1 and C2 may communicate with a public key server **805** located outside the network **1000**. The encryption server **105** may also communicate with the public key server **805** through communication channel **815**. Communication with the public key server **805** is made possible through a Translator program which may be stored on the encryption server **105**. First, a user may download the Enabler computer program **220** to client machine C1. The user may then execute the Enabler computer program **220** and transmit an encrypted digital message from client machine C1 to the public key server **805** through communication channels **810**. A user of client machine C2 may then retrieve the digital message from the public key server **805**, download the Enabler computer program **220** from the encryption server **105**, and decrypt the retrieved encrypted digital message.

FIG. **9** shows another embodiment of the public key cryptosystem with roaming user capability **200** where a user transmits and receives digital messages through a pop proxy **910**. This embodiment comprises an encryption server **105**, a client machine **110** containing a pop proxy **910**, and a pop server **905**. A user first downloads a pop proxy **910** application to his or her client machine **110**. The pop proxy **910** is then installed and configured to be the pop address that client machine **110** connects to. The pop proxy **910** is connected to and communicates directly with a pop account located on pop server **905** through communication channel **915**. The pop proxy **910** is also connected to the encryption server **105** through communication channels **115** and has access to both the New User computer program **215** and the Enabler computer program **220**.

Once the pop proxy **910** is installed and configured on the client machine **110**, the user may access a web page on the client machine **110** and download the New User computer program **215** to the pop proxy **910** from the encryption server **105**. The New user computer program **215** directs the client machine **110** to generate a user identifier **330**, a private key, and a public key **325**. The New User computer program **215** then encrypts the private key and transmits the encrypted private key **320** and public key **325** back to the encryption server **105**. The Server computer program **335** directs the encryption server **105** to receive the encrypted private key **320** and the public key **325** from the client machine **110** and store them in the encryption server **105** database **315**.

To transmit a digital message from the system depicted in FIG. **9**, the user first logs-in to the encryption server **105**. Here, the server is authenticated to the user by industry standard means, such as SSL using authentication certificates. For security purposes, a user may be limited to a certain number of log-in sessions per hour, such as forty, to prevent someone from trying to break into the network **1000** and obtain a user's encrypted private key **320**. The encryption server **105** then downloads the Enabler computer program **220** to the pop proxy **910**. The user then enters his or her passphrase, hashes the passphrase, and transmits the hashed passphrase to the encryption server **105**.

Next, the encryption server **105** authenticates the hashed passphrase and transmits the encrypted private key **320** back to the client computer **110**. The user may now decrypt the encrypted private key **320** with his or her passphrase, generate a digital message, and obtain a message recipient's public key from the pop server **905**. Finally, the user may encrypt the digital message with the recipient's public key, optionally signing the digital message with the client sender's private key, and transmit the encrypted digital message to the pop server **905**. All public keys of message recipients may be temporarily or permanently stored on the pop proxy **910** for speed in future message sending.

Once the encrypted digital message is stored on the pop server **905**, the pop server **905** client recipient to whom the encrypted digital message is directed may retrieve and decrypt the encrypted digital message with his private key. The digital message may be email, real-time chat, or any other form of digital message which may be transmitted over the network **1000**.

In another embodiment, the encrypted digital message does not have to be stored on the pop server **905**, but may instead be transmitted in any convenient way to the digital message recipient. For real time data that is time or bandwidth sensitive, (e.g., real time voice communication) encrypted digital message data may flow directly between both communicating client machines **110**.

In the process depicted in FIG. **6**, the user passphrase, plaint text private key, or encrypted private key **320** remains on the pop proxy **910** only for the duration of time in which the user is logged-in to the network **1000**. As soon as the user logs-off of the network **1000**, the passphrase is erased from the pop proxy **910**.

In another embodiment, the user passphrase, or private key may not be erased after logging-off the network **1000**. In this embodiment, the user passphrase or private key remain on the computer so the user rarely has to retype their passphrase or download the encrypted private key **320** from the encryption server **105**. The user passphrase or plain text private key, are never transmitted to the encryption server **105**.

FIG. **10** shows another embodiment of the public key cryptosystem with roaming user capability **200** where the network **1000** comprises multiple encryption servers **105** which all communicate with each other through communication channels **1010**. An example of an encryption server **105** may be a Sun Workstation, or a low cost personal computer operating on a Unix system. contain all or a subset of every user's encrypted private key **320**, public key **325**, user identifier **330**, or other user information. In this embodiment, the encryption server **105** administrator may have access to the private keys specific to each encryption server **105** on the network **1000**.

Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the spirit and scope of the invention as described by the appended claims.

What is claimed is:

1. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and wherein said network comprises a plurality of client computers and encryption servers, further wherein each encryption server can communicate with every other encryption server on said network.

2. The system of claim **1**, wherein said client computer is further operable to store and access a New User computer program, said client computer processor operable to execute said New User computer program such that said client computer is directed by said New User computer program to communicate with said Server computer program to:

generate said public/private key pair;

generate said user passphrase;

generate said user identifier;

hash said user passphrase;

transmit said hash of said user passphrase to said encryption server to compare against a plurality of hashed English words, common nouns, and popular sayings located on said database of said encryption server;

encrypt said private key with said hash of said user passphrase yielding said encrypted private key; and

transmit said encrypted private key and said public key to said encryption server.

3. The system of claim **1**, wherein said user identifier is a user log-in ID or said user passphrase, and further wherein said user log-in ID or user passphrase is hashed and transmitted to said encryption server and compared against said database of hashed user identifiers to verify the identity of said user.

4. The system of claim **2**, wherein said encryption server is further operable to execute said Server computer program to communicate with said New User computer program such that said encryption server is directed by said Server computer program to:

receive and compare said hash of said passphrase against a plurality of hashed English words, common nouns, and popular sayings located on said database of said encryption server;

compare said hash of said passphrase against said database of hashed passphrases to verify the identity of said user;

receive said encrypted private key and said public key paired to said encrypted private key from said client computer; and

store said encrypted private key and said public key in said database of said encryption server.

5. The system of claim **1**, wherein said Enabler computer program is further executable to transmit other user specific information from said client computer to said encryption server, said Server computer program is further executable to transmit other user specific information from said encryption server database to said client computer.

6. The system of claim **1**, wherein said user may use any client computer on said network to access said encrypted private key, thus giving said user roaming capability.

7. The system of claim **1**, wherein said user passphrase remains on said client computer for the duration of time said user is logged-in to said encryption server, further wherein said user passphrase is never transmitted to said encryption server and is erased from said client computer when said user logs-off said network.

8. The system of claim **1**, wherein said user passphrase or private key may not be erased after logging-off said network, said user passphrase or said private key remain on said computer.

9. The system of claim **1**, wherein said encrypted digital message resides on said encryption server and may not be accessed by anyone but an intended user recipient, further wherein said digital message may be in the form of email or real-time chat.

10. The system of claim **1**, wherein a secure socket layer exists between said client sender computer and said encryption server, and wherein said secure socket layer also exists between said encryption server and said client recipient computer.

11. The system of claim **2**, wherein said New User computer program and said Enabler computer program are downloaded from said encryption server or are directly installed on said client computer.

12. The system of claim **1**, wherein said private key is symmetrically encrypted with said passphrase and stored on either said encryption server or said client computer.

13. The system of claim **1**, wherein said encryption server allows a limited number of log-on attempts.

14. The system of claim 1, wherein said digital message is encrypted using any public/private key cipher including RSA, Elliptical Curve, or Diffie-Helman.

15. The system of claim 1, wherein said encrypted digital message is transmitted from said client sender computer to a server outside said network, then from said server outside said network to said client recipient computer.

16. The system of claim 1, wherein each encryption server on said network contains all or a subset of every user's encrypted private key, public key, user identifier, or other user information.

17. The system of claim 16, wherein each encryption server of said network has its own public/private key pair, further wherein each encryption server has access to said public/private key pairs of every other encryption server on said network.

18. The system of claim 17, wherein only an encryption server administrator has access to said private keys of each encryption server on said network.

19. The system of claim 1, wherein a cyclic redundancy check (CRC) is added to the end of said digital message before encrypting it.

20. The system of claim 1, wherein said encryption server includes a translator computer program to communicate with other public/private key encryption servers operating under a different standard certificate of authority.

21. The system of claim 1, wherein said public keys are stored on said encryption server in plain text form.

22. The system of claim 1, wherein said Server computer program and said New User computer program are divided into two or more subprograms.

23. The system of claim 1, wherein said user passphrase is generated by said New User computer program.

24. The system of claim 1, wherein said passphrase is actively or passively generated by true random numbers.

25. The system of claim 1, wherein said encryption server is authenticated to said user by industry standard means, such as SSL, using authentication certificates.

26. The system of claim 1, wherein said user may optionally sign said digital message with said private key before encrypting and transmitting said digital message to said encryption server.

27. The system of claim 1, wherein said digital message contains time or bandwidth sensitive data, and wherein said digital message need not be transmitted through said encryption server, and further wherein said time or bandwidth sensitive data is encrypted and transmitted directly to said client recipient computer.

28. The system of claim 1, wherein said passphrase, private key, or said user recipient's public is not erased after logging-off said network, and said passphrase, said private key, or said user recipient public key remains on said computer.

29. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

at said client sender machine:

entering a user identifier; and

transmitting said user identifier to an encryption server;

at said encryption server:

receiving said user identifier;

comparing said user identifier against a plurality of user identifiers located in a database on said encryption server to verify the identity of said user;

retrieving a private key encrypted with a passphrase from said database of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair; and

transmitting said encrypted private key from said encryption server to said user's client machine;

at said client sender machine:

receiving said encrypted private key from said encryption server;

decrypting said encrypted private key with said passphrase;

generating a digital message;

retrieving a user recipient's public key from said encryption server database;

encrypting said digital message with said user recipient's public key; and

transmitting said encrypted digital message to said client recipient machine; and wherein said method employs a network comprised of a plurality of client computers and encryption servers, further wherein each encryption server can communicate with every other encryption server on said network.

30. The method of claim 29, further comprising the following steps prior to entering said user identifier,

at said client sender machine:

generating said public/private key pair;

generating said user passphrase;

generating said user identifier, wherein said identifier can be a user log-in ID;

hashing said user passphrase;

transmitting said hash of said user passphrase to said encryption server to compare against said database of hashed English words, common nouns, and popular sayings;

encrypting said private key with said hash of said user passphrase yielding said encrypted private key; and

transmitting said encrypted private key and said public key to said encryption server;

at said encryption server:

receiving said encrypted private key and said public key; and

storing said encrypted private key and said public key in said database of said encryption server.

31. The method of claim 29, wherein said user identifier is said user's passphrase, further wherein said user's passphrase is hashed and transmitted to said encryption server and compared against said database of hashed passphrases to verify the identity of said user.

32. The method of claim 29, wherein said encrypted digital message is transmitted from said client sender machine to said encryption server, then transmitted from said encryption server to said client recipient machine.

33. The method of claim 29, wherein said encrypted digital message is transmitted from said client sender machine to a server outside said network then from said server outside said network to said client recipient machine.

34. The method of claim 29, wherein said passphrase is actively or passively generated by true random numbers.

35. The method of claim 29, wherein said user may optionally sign said digital message with said private key before encrypting and transmitting said digital message to said encryption server.

36. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

entering a user identifier; and

transmitting said user identifier to an encryption server to verify identity of said user; and

downloading an Enabler computer program from said encryption server to said client sender's machine,

wherein said Enabler computer program is executable to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key from said encryption server database;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient; and wherein said method employs a network comprised of a plurality of client computers and encryption servers, further wherein each encryption server can communicate with every other encryption server on said network.

37. The method of claim **36**, wherein a New User computer program is downloaded from said encryption server to said client seder's machine, further wherein said New User computer program is executable to communicate with a Server computer program located on said encryption server to:

generate said public/private key pair;

generate said user passphrase;

generate said user identifier;

hash said user passphrase;

transmit said hash of said user passphrase to said encryption server to compare against a plurality of hashed English words, common nouns, and popular sayings located on said database of said encryption server;

encrypt said private key with said hash of said user passphrase yielding said encrypted private key; and

transmit said encrypted private key and public key to said encryption server.

38. The method of claim **36**, wherein said user identifier is said user's passphrase, further wherein said user's passphrase is hashed and transmitted to said encryption server and compared against said database of hashed passphrases to verify the identity of said user.

39. The method of claim **36**, wherein said New User computer program and said Enabler computer program are directly loaded onto said client sender's machine.

40. The method of claim **36**, wherein logging into said encryption server comprises the steps of finding a log-in web page for said encryption server on the Internet and typing in a user's identifier.

41. The method of claim **36**, wherein said encrypted digital message is transmitted from said client sender machine to said encryption server, then transmitted from said encryption server to said client recipient machine.

42. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said user passphrase remains on said client server computer for the duration of time said user is logged-in to said encryption server,

further wherein said user passphrase is never transmitted to said encryption server and is erased from said client computer when said user logs-off said network.

43. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said user passphrase or private key may not be erased after logging-off said network.

44. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said encrypted digital message resides on said encryption server and may not be accessed by anyone but an intended user recipient.

45. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and wherein a secure socket layer exists between said client sender computer and said encryption server, and

wherein said secure socket layer also exists between said encryption server and said client recipient computer.

46. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said encryption server allows a limited number of log-on attempts.

47. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said encrypted digital message is transmitted from said client sender computer to a server outside said network, then from said server outside said network to said client recipient computer.

48. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein a cyclic redundancy check (CRC) is added to the end of said digital message before encrypting it.

49. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said encryption server is authenticated to said user by industry standard means, such as SSL, using authentication certificates.

50. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said user may optionally sign said digital message with said private key before encrypting and transmitting said digital message to said encryption server.

51. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said digital message contains time or bandwidth sensitive data, and wherein said digital message need not be transmitted through said encryption server, and further wherein said time or bandwidth sensitive data is encrypted and transmitted directly to said client recipient computer.

52. A system for sending an encrypted digital message from a user at a client sender computer to a client recipient computer over a network, comprising:

a client computer operable to access an Enabler computer program, said client computer comprising:

a client memory operable to store said Enabler computer program;

a client processor electrically connected to said client memory, said client processor operable to execute said Enabler computer program such that said client computer is directed by said Enabler computer program to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said

encryption server, said private key having a corresponding public key forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient;

an encryption server, said encryption server operable to process requests from said client computer, said encryption server comprising:

a server memory operable to store said Server computer program and a database, said database comprising a plurality of said user identifiers, encrypted private keys, and public keys; and

a server processor electronically connected to said server memory, said server processor operable to execute said Server computer program such that said encryption server is directed by said Server computer program to communicate with said Enabler computer program to:

receive and compare said user identifier against a plurality of user identifiers located in said database of said encryption server to verify identity of said user;

retrieve said encrypted private key from said encryption server database; and

transmit said encrypted private key from said encryption server to said user's client computer; and

a network comprising said client sender computer, said encryption server, and said client recipient computer, wherein said network allows communication between said client sender computer and said encryption server and further between said encryption server and said client recipient computer; and

wherein said passphrase, private key, or said user recipient's public key is not erased after logging-off said network, and said passphrase, said private key, or said user recipient public key remains on said computer.

53. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

at said client sender machine:

entering a user identifier; and

transmitting said user identifier to an encryption server;

at said encryption server:

receiving said user identifier;

comparing said user identifier against a plurality of user identifiers located in a database on said encryption server to verify the identity of said user;

retrieving a private key encrypted with a passphrase from said database of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair; and

transmitting said encrypted private key from said encryption server to said user's client machine;

at said client sender machine:

receiving said encrypted private key from said encryption server;

decrypting said encrypted private key with said passphrase:

generating a digital message;

retrieving a user recipient's public key from said encryption server database;

encrypting said digital message with said user recipient's public key; and

transmitting said encrypted digital message to said client recipient machine; and

wherein said user identifier is said user's passphrase, further wherein said user's passphrase is hashed and transmitted to said encryption server and compared against said database of hashed passphrases to verify the identity of said user.

54. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

at said client sender machine:

entering a user identifier; and

transmitting said user identifier to an encryption server;

at said encryption server:

receiving said user identifier;

comparing said user identifier against a plurality of user identifiers located in a database on said encryption server to verify the identity of said user;

retrieving a private key encrypted with a passphrase from said database of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair; and

transmitting said encrypted private key from said encryption server to said user's client machine;

at said client sender machine;

receiving said encrypted private key from said encryption server;

decrypting said encrypted private key with said passphrase;

generating a digital message;

retrieving a user recipient's public key from said encryption server database;

encrypting said digital message with said user recipient's public key; and

transmitting said encrypted digital message to said client recipient machine; and

wherein said user encrypted digital message is transmitted from said client sender machine to said encryption server, then transmitted from said encryption server to said client recipient machine.

55. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

at said client sender machine:

entering a user identifier; and

transmitting said user identifier to an encryption server;

at said encryption server;

receiving said user identifier;

comparing said user identifier against a plurality of user identifiers located in a database on said encryption server to verify the identity of said user;

retrieving a private key encrypted with a passphrase from said database of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair; and

transmitting said encrypted private key from said encryption server to said user's client machine;

at said client sender machine:

receiving said encrypted private key from said encryption server;

decrypting said encrypted private key with said passphrase;

generating a digital message;

retrieving a user recipient's public key from said encryption server database;

encrypting said digital message with said user recipient's public key; and

transmitting said encrypted digital message to said client recipient machine; and

wherein said encrypted digital message is transmitted from said client sender machine to a server outside said network then from said server outside said network to said client recipient machine.

56. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

at said client sender machine:

entering a user identifier; and

transmitting said user identifier to an encryption server;

at said encryption server:

receiving said user identifier;

comparing said user identifier against a plurality of user identifiers located in a database on said encryption server to verify the identity of said user;

retrieving a private key encrypted with a passphrase from said database of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair; and

transmitting said encrypted private key from said encryption server to said user's client machine;

at said client sender machine:

receiving said encrypted private key from said encryption server;

decrypting said encrypted private key with said passphrase;

generating a digital message;

retrieving a user recipient's public key from said encryption server database;

encrypting said digital message with said user recipient's public key; and

transmitting said encrypted digital message to said client recipient machine; and

wherein said user may optionally sign said digital message with said private key before encrypting and transmitting said digital message to said encryption server.

57. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

entering a user identifier; and

transmitting said user identifier to an encryption server to verify identity of said user; and

downloading an Enabler computer program from said encryption server to said client sender's machine, wherein said Enabler computer program is executable to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key from said encryption server database;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient; and

wherein a New User computer program is downloaded from said encryption server to said client sender's machine, further wherein said New User computer program is executable to communicate with a Server computer program located on said encryption server to:

generate said public/private key pair;

generate said user passphrase;

generate said user identifier;

hash said user passphrase;

transmit said hash of said user passphrase to said encryption server to compare against a plurality of hashed English words, common nouns, and popular sayings located on said database of said encryption server;

encrypt said private key with said hash of said user passphrase yielding said encrypted private key; and

transmit said encrypted private key and public key to said encryption server.

58. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

entering a user identifier; and

transmitting said user identifier to an encryption server to verify identity of said user; and

downloading an Enabler computer program from said encryption server to said client sender's machine, wherein said Enabler computer program is executable to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key from said encryption server database;

encrypt a digital message with said user recipient's public key; and

transmit said encrypted digital message to said user recipient; and

wherein said New User computer program and said Enabler computer program are directly loaded onto said client sender's machine.

59. A method for sending an encrypted digital message from a client sender machine to a client recipient machine comprising the steps of:

entering a user identifier; and

transmitting said user identifier to an encryption server to verify identity of said user; and

downloading an Enabler computer program from said encryption server to said client sender's machine, wherein said Enabler computer program is executable to communicate with a Server computer program located on said encryption server to:

allow said user to enter a user identifier;

transmit said user identifier to said encryption server to verify identity of said user;

receive a private key encrypted with a passphrase from a database located in a memory of said encryption server, said private key having a corresponding public key, thereby forming a public/private key pair;

use said passphrase to decrypt said encrypted private key at said client computer;

retrieve a user recipient's public key from said encryption server database;

encrypt a digital message with said user recipient's public key; and transmit said encrypted digital message to said user recipient; and

wherein said encrypted digital message is transmitted from said client sender machine to said encryption server, then transmitted from said encryption server to said client recipient machine.

* * * * *

US006178507B1

(12) **United States Patent**
Vanstone

(10) **Patent No.: US 6,178,507 B1**
(45) **Date of Patent: Jan. 23, 2001**

(54) **DATA CARD VERIFICATION SYSTEM**

(75) Inventor: **Scott A Vanstone**, Waterloo (CA)

(73) Assignee: **Certicom Corp.**, Ontario (CA)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **09/016,926**

(22) Filed: **Feb. 2, 1998**

(30) **Foreign Application Priority Data**

Feb. 3, 1997 (GB) 9702152

(51) **Int. Cl.**[7] **H04L 9/00**

(52) **U.S. Cl.** **713/169**; 380/43; 380/259; 380/283; 380/285; 705/67; 713/168; 713/172; 713/176; 713/180

(58) **Field of Search** 380/21, 23, 24, 380/30, 44, 283, 285; 705/64, 65, 67, 71, 73; 713/168, 169, 172, 176, 180

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,748,668		5/1988	Shamir et al.	380/29
4,890,323		12/1989	Beker et al.	380/28
4,995,082	*	2/1991	Schnorr	380/23
5,218,637		6/1993	Angebaud et al.	380/25
5,299,263	*	3/1994	Beller et al.	380/30
5,400,403		3/1995	Fahn et al.	380/21
5,406,628	*	4/1995	Beller et al.	380/30
5,627,893		5/1997	Demytko	380/30
5,721,781	*	2/1998	Deo et al.	380/25
5,748,740	*	5/1998	Curry et al.	380/25
5,793,866		8/1998	Brown et al.	380/2
5,805,702	*	9/1998	Curry et al.	380/24
5,825,880		10/1998	Sudia et al.	380/21
5,870,470		2/1999	Johnson et al.	380/6
5,881,038		3/1999	Oshima et al.	369/59
5,907,618		5/1999	Gennaro et al.	380/21
5,917,913		6/1999	Wang	380/25
5,955,717		9/1999	Vanstone	235/380
5,960,084		9/1999	Angelo	380/25
6,038,549		3/2000	Davis et al.	705/35
6,041,314		3/2000	Davis	705/41

FOREIGN PATENT DOCUMENTS

0 588 339	3/1994	(EP) .
2 536 928	6/1984	(FR) .
WO 91/16691	10/1991	(WO) .

OTHER PUBLICATIONS

Schneier, Bruce, Applied Cryptography, 1996, pp.35–36.
Miyaji, A: "Elliptic Curves Suitable For Cryptosystem", IEICE Transactions On Fundamentals of Electronics, Communications and Computer Sciences, vol. E77–A, No.1, Jan. 1, 1994, pp. 98–104, XP000439669.
Schnorr, C P: "Efficcient Signature Generation By Smart Cards", Journal of Cryptology, vol. 4, No. 3, Jan. 1, 1991, pp. 161–174, XP000574352.
Kenji, Koyoma et al: "Elliptic Curve Cryptosystems And Their Applications", IEICE Transactions On Information And Systems, vol. E75–D, No. 1, Jan. 1, 1992, pp. 50–57, XP000301174.
Waleffe, D De et al: "Corsair: A Smart Card For Public Key Cryptosystems", Advances In Cryptology—Proceedings of Crypto, Santa Barbara, Aug. 11–15, 1990, No. CONF. 10, Jan. 1, 1990, pp. 502–513, XP000260013, Menezes, A J; Vanstone, S A.
Koblitz, N: "Elliptic Curve Cryptosystems", Mathematics Of Computation, vol. 48, No. 177, Jan. 1987, pp. 203–209, XP000671098.

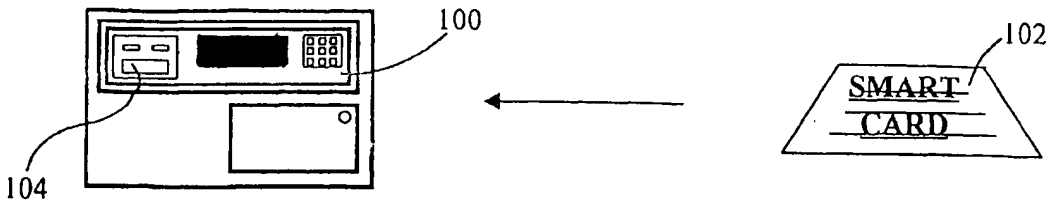
* cited by examiner

Primary Examiner—Tod R. Swann
Assistant Examiner—Paul E. Callahan
(74) *Attorney, Agent, or Firm*—Finnegan, Henderson, Farabow, Garrett, & Dunner, L.L.P.

(57) **ABSTRACT**

A method for verifying the authenticity of messages exchanged between a pair of correspondents in an electronic conducted over a data transmission system where the correspondents each include respective signing and verifying portions of a first signature scheme and a second signature scheme different from the first and utilizing an elliptic curve cryptosystem.

8 Claims, 2 Drawing Sheets



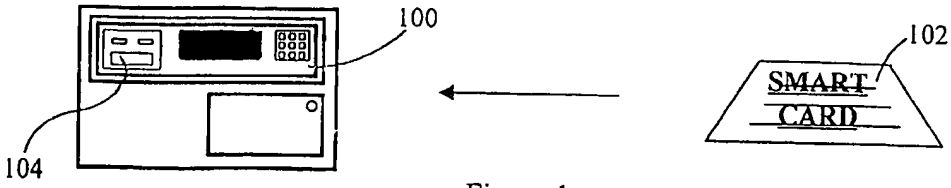


Figure 1a

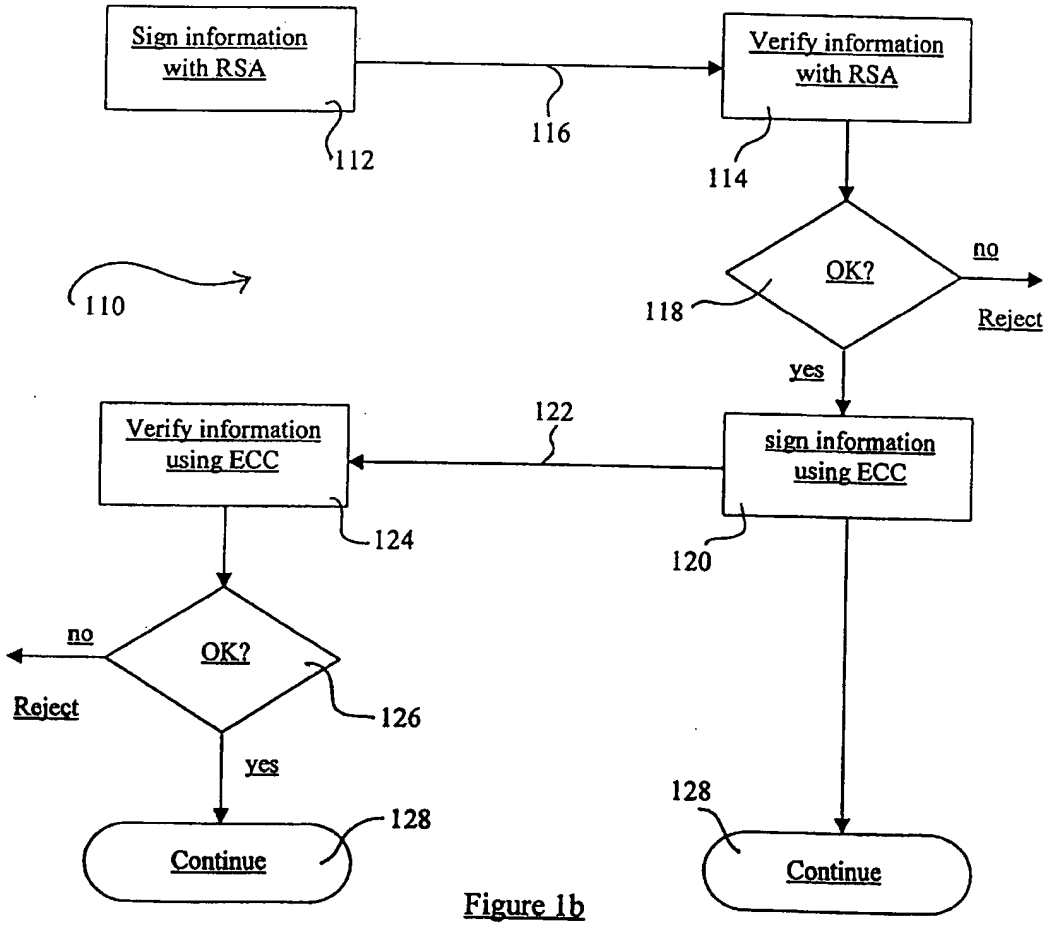


Figure 1b

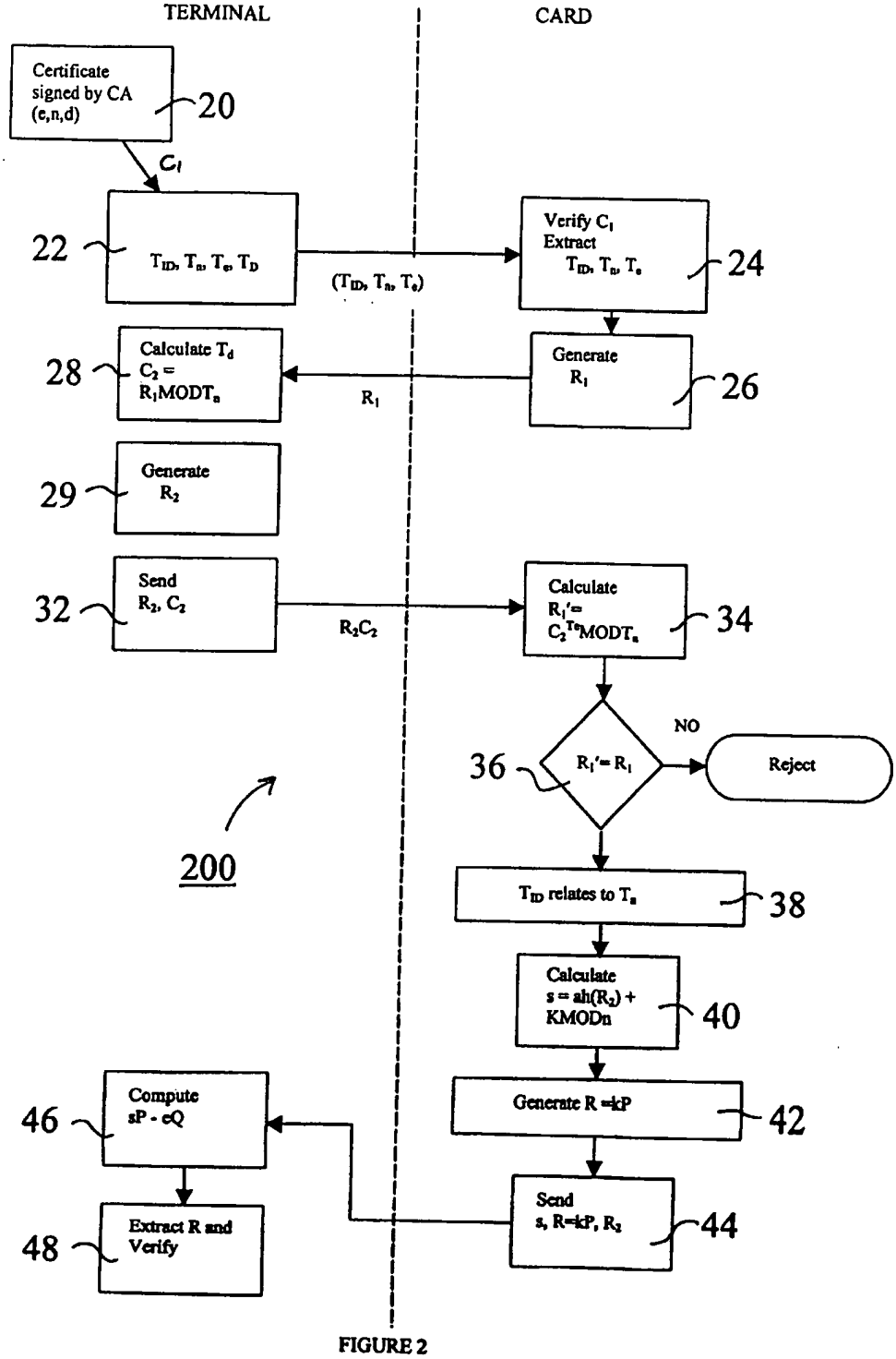

TERMINAL
CARD
Certificate signed by CA (e,n,d)
20
C1
22
TID, Tn, Te, TD
(TID, Tn, Te)
Verify C1 Extract TID, Tn, Te
24
Generate R1
26
R1
28
Calculate Td C2 = R1MODTn
29
Generate R2
32
Send R2, C2
R2C2
Calculate R1' = C2^TeMODTn
34
NO
36
R1' = R1
Reject
200
TID relates to Tn
38
Calculate s = sh(R2) + KMODn
40
Generate R =kP
42
Send s, R=kP, R2
44
Compute sP - eQ
46
Extract R and Verify
48

FIGURE 2

DATA CARD VERIFICATION SYSTEM

This invention relates to methods and apparatus for data transfer and authentication in an electronic transaction system, and more particularly to electronic transaction systems utilizing smart cards.

BACKGROUND OF THE INVENTION

It has become widely accepted to conduct transactions such as financial transactions or exchange of documents electronically. Automated teller machines (ATMs) and credit cards are widely used for personal transaction and as their use expands so too does the need to verify such transactions increase. A smart card is somewhat like a credit card and includes some processing and storage capability. Smart cards are prone to fraudulent misuse, for example by a dummy terminal which is used to glean information from an unsuspecting user. Thus, before any exchange of critical information takes place between either a terminal and a smart card or vice versa it is necessary to verify the authenticity of the terminal as well as the card. One of these verifications may take the form of "signing" an initial transaction digitally so that the authenticity of the transaction can be verified by both parties involved in the subsequent session. The signature is performed according to a protocol that utilizes a random message, i.e. the transaction and a secret key associated with the party.

The signature must be performed such that the party's secret key cannot be determined. To avoid the complexity of distributing secret keys, it is convenient to utilize a public key encryption scheme in the generation of the signature. Such capabilities are available where the transaction is conducted between parties having access to relatively large computing resources, but it is equally important to facilitate such transactions at an individual level where more limited computing resources available, as in the smart card.

Transaction cards or smart cards are now available with limited computing capacity, but these are not sufficient to implement existing digital signature protocols in a commercially viable manner. As noted above, in order to generate a verification signature it is necessary to utilize a public key inscription scheme. Currently, most public key schemes are based on RSA, but the DSS and the demand for a more compact system are rapidly changing this. The DSS scheme, which is an implementation of a Diffie-Hellman public key protocol, utilizes the set of integers Z_p where p is a large prime. For adequate security, p must be in the order of 512 bits, although the resultant signature may be reduced mod q, where q divides p−1, and may be in the order of 160 bits.

An alternative encryption scheme which was one of the first fully fledged public key algorithms and which works for encryption as well as for digital signatures is known as the RSA algorithm. RSA gets it security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) of prime numbers. The public key for RSA encryption is n, the product of the two primes p and q where p and q must remain secret and e which is relatively prime to $(p-1)\times(q-1)$. the encryption key d is equal to $e^{-1} \pmod{(p-1)\times(q-1)}$. Note that d and n are relatively prime.

To encrypt a message m, first divide into a number of numerical blocks such that each block is a unique representation modulo n, then the encrypted message block c_i is simply $m_i^c \pmod{n}$. To decrypt a message take each encrypted block c_i and compute $m_i = c_i^d \pmod{n}$.

Another encryption scheme that provides enhanced security at relatively small modulus is that utilizing elliptic curves in the finite field 2^m. A value of m in the order of 155 provides security comparable to a 512 bit modulus DSS and therefore offers significant benefits in implementation.

Diffie-Hellman public key encryption utilizes the properties of discrete logs so that even if a generator β and the exponentiation β^k is known, the value of k cannot be determined. A similar property exist with elliptic curves where the addition of two points on any curve produces a third point on the curve. Similarly, multiplying a point P on the curve by an integer k produces a further point on the curve. For an elliptic curve, the point kP is simply obtained by adding k copies of the point P together.

However, knowing the starting point and the end point does not reveal the value of the integer k which may then be used as a session key for encryption. The value kP, where P is an initial known point is therefore equivalent to the exponentiation β^k. Furthermore, elliptic curve crypto-systems offer advantages over other key crypto-systems when bandwidth efficiency, reduced computation and minimized code space are application goals.

Furthermore, in the context of a smart card and an automated teller machine transaction, there are two major steps involved in the authentication of both parties. The first is the authentication of the terminal by the smart card and the second is the authentication of the smart card by the terminal. Generally, this authentication involves the verification of a certificate generated by the terminal and received by the smart card and the verification of a certificate signed by the smart card and verified by the terminal. Once the certificates have been positively verified the transaction between the smart card and the terminal may continue.

Given the limited processing capability of the smart card, verifications and signature processing performed on the smart card are generally limited to simple encryption algorithms. A more sophisticated encryption algorithm is generally beyond the scope of the processing capabilities contained within the smart card. Thus, there exist a need for a signature verification and generation method which may be implemented on a smart card and which is relatively secure.

SUMMARY OF THE INVENTION

This invention seeks in one aspect to provide a method of data verification between a smart card and a terminal.

In accordance with this aspect there is provided a method for verifying a pair of participants in an electronic transaction, comprising the steps of verifying information received by the second participant from the first participant, wherein the verification is performed according to a first signature algorithm; verifying information received by the first participant from the second participant, wherein the verification is performed according to a second signature algorithm; and whereby the transaction is rejected if either verification fails.

The first signature algorithm may be one which is computationally more difficult in signing than verifying, while the second signature algorithm is more difficult in verifying than signing. In such an embodiment the second participant may participate with relatively little computing power, while security is maintained at a high level.

In a further embodiment, the first signature algorithm is based on an RSA, or DDS type algorithm, and the second signature algorithm is based on an elliptic curve algorithm.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described by way of example on the reference to the accompanying drawings, in which,

FIG. 1*a* is a schematic representations showing a smart card and terminal;

FIG. 1*b* is a schematic representations showing the sequence of events performed during the verification process in a smart card transaction system; and

FIG. 2 is a detailed schematic representation showing a specific protocol.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Referring to FIG. 1(*a*), a terminal **100** is adapted to receive a smart card **102**. Typically, insertion of the card **102** into the terminal initiates a transaction. Mutual authentication between the terminal and the card is then performed as shown in FIG. 1*b*. In very general terms, this mutual authentication is performed according to a "challenge-response" protocol. Generally, card transmits information to the terminal, the terminal **100** signs information with an RSA based algorithm **112** and is then sent to the card **102**, which verifies the information with an RSA based algorithm **114**. The information exchange **116** between the card and the terminal also includes information generated by the card which is sent to the terminal to be signed by the terminal with an RSA algorithm and returned to the card to be verified utilizing a RSA algorithm. Once the relevant verification has been performed **118**, a further step is performed where information is signed by the card using an elliptic curve encryption protocol **120** and submitted to the terminal to be verified **124** by the terminal utilizing an elliptic curve based protocol. Similarly, the information exchange **122** between the card and the terminal may include information generated by the terminal which is sent to the card to be signed by the card and returned to the terminal for verification. Once the appropriate information has been verified **126** the further transactions between the terminal and card may proceed **128**.

Referring now to FIG. **2**, a detailed implementation of the mutual authentication of the terminal and the card, according to the "challenged-response" protocol is shown generally by numeral **200**. The terminal **100** is first verified by the card **102** and the card is then verified by the terminal. The terminal first sends to the card a certificate C_1, **20** containing its ID, T_{ID}, and public information including the public key. The certificate **20** may be also signed by a certifying authority (CA) so that the card may verify the association of the terminal ID T_{ID} with the public key received from the terminal. The keys used by the terminal and the CA in this embodiment may both be based on the RSA algorithm.

With the RSA algorithm each member or party has a public and a private key, and each key has two parts. The signature has the form:

$$S=m^d(mod\ n)$$

where:

m is the message to be signed;

n a public key is the modulus and is the product of two primes p and q;

e the encryption key chosen at random and which is also public is a number chosen to be relatively prime to $(p-1)\times(q-1)$; and

d the private key which is congruent to e^{-1} $(\text{mod}(p-1)\times(q-1))$.

For the RSA algorithm, the pair of integers (n,e) are the public key information that is used for signing. While, the pair of integers (d,n) may be used to decrypt a message which has been encrypted with the public key information (n,e).

Referring back to FIG. **2**, the numbers n and e are the public keys of the CA and may be set as system parameters. The public key e may be either stored in the smart card or in an alternate embodiment hardwired into an logic circuit in the card. Furthermore, by choosing e to be relatively small, ensures that the exponentiation may be carried out relatively quickly.

The certificate **20** C_1 is signed by the CA and has the parameters (n,e). The certificate contains the terminal ID T_{Id} and the terminal public key information T_n and T_e which is based on the RSA algorithm. The certificate C_1 is verified **24** by the card extracting T_{ID}, T_n, T_e. This information is simply extracted by performing C_1^e mod n. The card then authenticates the terminal by generating a random number R1, **26**, which it transmits to the terminal. The terminal signs the message R1 using its secret key T_d by performing $R1^{Te}$ $MODT_n$ to generate the value C_2, **28**. Once again the key used by the terminal is an RSA key which has been originally created in such a way that the public key T_e consist of a small possibly system wide parameter having a value 3, while the other part of the public key is the modulus T_n which would be associated with the terminal. The terminals private key T_d cannot be small if it corresponds to a small public key T_e. In the case of the terminal, it does not matter whether the private key T_d is chosen to be large as the terminal has the required computing power to perform the exponentiation relative quickly.

Once the terminal has calculated the value C_2, **28**, it generates a secret random number R2, **29** the terminal sends both R2 and C_2, **32** to the card. The card then performs a modular exponentiation **34** on the signed value C_2 with the small exponent T_e, using the terminal's modulus T_n. This is performed by calculating $R1'=C_2^{Te}$ mod T_n. If R1' is equal to R1, **36** then the card knows that it is dealing with the terminal whose ID T_{ID} is associated **38** with the modulus T_n. The card generally contains a modulo arithmetic processor (not shown) to perform the above operation.

The secret random number R2 is signed **40** by the card and returned to the terminal along with a certificate signed by the CA which relates the card ID to its public information. The signing by the card is performed according to an elliptic curve signature algorithm.

The verification of the card proceeds on a similar basis as the verification of the terminal, however, the signing by the card utilizes an elliptic curve encryption system.

Typically for an elliptic curve implementation a signature component s has the form:

$$s=ae+k(mod\ n)$$

where:

P is a point on the curve which is a predefined parameter of the system;

k is a random integer selected as a short term private or session key, and has a corresponding short term public key R=kP;

a is the long term private key of the sender (card) and has a corresponding public key aP=Q;

e is a secure hash, such as the SHA hash function, of a message m (R2 in this case) and short term public key R; and

n is the order of the curve.

For simplicity it will be assumed that the signature component s is of the form s=ae+k as discussed above although it will be understood that other signature protocols may be used.

To verify the signature sP−eQ must be computed and compared with R. The card generates R, using for example a field arithmetic processor (not shown). The card sends to

the terminal a message including m, s, and R, indicated in block **44** of FIG. **2** and the signature is verified by the terminal by computing the value (sP−eQ) **46** which should correspond to kP. If the computed values correspond **48** then the signature is verified and hence the card is verified and the transaction may continue.

The terminal checks the certificate, then it checks the signature of the transaction data which contains R2, thus authenticating the card to the terminal. In the present embodiment the signature generated by the card is an elliptic curve signature, which is easier for the card to generate, but requires more computation by the terminal to verify.

As is seen from the above equation, the calculation of s is relatively straightforward and does not require significant computing power. However in order to perform the verification it is necessary to compute a number of point multiplications to obtain sP and eQ, each of which is computationally complex. Other protocols, such as the MQV protocols require similar computations when implemented over elliptic curves which may result in slow verification when the computing power is limited. However this is generally not the case for a terminal.

Although an embodiment of the invention has been described with reference to a specific protocol for the verification of the terminal and for the verification of the card, other protocols may also be used.

What is claimed is:

1. A method of verifying the authenticity of messages exchanged between a pair of correspondents in an electronic transaction conducted over a data transmission system, said correspondents each including respective signing and verifying portions of a first signature scheme and a second signature scheme different to said first scheme and utilizing an elliptic curve crypto system said method comprising the steps of:

one of said correspondents signing a message according to a signing portion of one of said schemes associated with said one correspondent to provide a first signed message and transmitting said first signed message to another of said correspondents; said other correspondent utilizing said verifying portion of said one signature scheme to verify said first signed message received from said one correspondent;

said other correspondent signing a message by utilizing said signing portion of the other of said signature schemes to provide a second signed message and transmitting a second signed message to said one correspondent;

said one correspondent verifying said second signed message received from said other correspondent by utilizing said verification portion of said other of said signature schemes, wherein one of said signature and one of said verifications is performed according to said second signature scheme utilizing an elliptic curve cryptosystem; and rejecting said transaction if either verification fails.

2. A method as defined in claim **1**, said first signature scheme is computationally more difficult in signing than verifying, while said second signature scheme is computationally more difficult in verifying than signing, thereby allowing one of said correspondents to participate with relatively little computing power while maintaining security of said transaction.

3. A method as defined in claim **1**, wherein said first digital signature scheme is an RSA type scheme.

4. A method as defined in claim **1**, wherein said first digital signature scheme is a DSS type scheme.

5. A method of verifying the authenticity of messages exchanged between a pair of correspondents in electronic transaction conducted over a data transmission system, said correspondents each including respective signing and verifying portions of a first signature scheme and a second signature scheme, different from said first scheme and utilizing an elliptic curve crypto system said method comprising the steps of:

one of said correspondents transmitting to another of said correspondents, a first certificate including public key and identification information of said first correspondent;

said other correspondent verifying said certificate and extracting said public key said identification information therefrom;

said other correspondent generating a first challenge R_1 and transmitting said challenge to said one correspondent;

said one correspondent signing said received challenge R_1 in accordance with said signing portion of one of said signature schemes to provide a second certificate C2;

said one correspondent generating a second challenge and transmitting said second challenge along with said certificate C2 to said other correspondent;

said other correspondent verifying said certificate C2 in accordance with said verification portion of one of said signature schemes;

said other correspondent signing said second challenge R2 in accordance with said signing portion of the other of said signature schemes to provide a third certificate and transmitting said said third certificate to said one correspondent; and

said one correspondent verifying said third certificate in accordance with said verification portion of said other of said signature schemes, and rejecting said transaction if either said signature is not verified.

6. A smart card for use in an electronic transaction with a correspondent, said card comprising:

a memory including

a verification algorithm of a first signature scheme to implement a verification of a signature performed according to a first signature generation algorithm by said correspondent;

a signing algorithm of second signature scheme different to said first signature scheme and utilizing elliptic curve cryptography, said algorithm implementing a signature according to a second signature generation algorithm;

a program for invoking said algorithms; and

processor means for running said first verification algorithm for verifying a first message signed by sad correspondent and for running said second signature for signing a second message for transmission to said correspondent.

7. A card according to claim **6** wherein said verification algorithm verifies an RSA signature.

8. A card according to claim **6** wherein said verification algorithm verifies a DSS signature.

* * * * *